

Universidad Carlos III de Madrid

Escuela politécnica superior



Grado en Ingeniería Informática

Trabajo Fin de Grado

**Localización y recogida de
objetos presentes en el entorno
con el robot NAO**

Autor: David Rodríguez Reyes

Tutor: Moises Martinez Muñoz

Resumen

En este trabajo se presenta el desarrollo de un sistema mediante el cual el robot humanoide NAO es capaz de localizar y recoger un objeto que se encuentre en su entorno. Para llevar a cabo esta tarea se ha desarrollado un módulo de visión artificial con el fin de detectar un determinado objeto y calcular la distancia al mismo. Mediante este sistema el robot será capaz de calcular los diferentes movimientos necesarios para acercarse al objeto y recogerlo haciendo uso de la manos. Para comprobar el correcto funcionamiento del sistema se realizarán una serie de pruebas mediante las cuales el robot NAO deberá recoger un objeto en diferentes situaciones.

Abstract

In this paper is presented the development of a system where the NAO humanoid robot is able to locate and pick up an object that is in environment. This task has developed using a computer vision module to detect an object and calculate the distance to it. Through this system, the robot will be able to calculate the movements necessary to approach the object and picked using the hands. The system has been tested using a set of tests where the NAO robot must pick up an object in different situations.

Índice general

1. Introducción	1
1.1. Descripción del problema	1
1.2. Motivación	2
1.3. Objetivos del trabajo	3
1.4. Estructura del documento	5
2. Estado de la Cuestión	7
2.1. Robótica	7
2.1.1. Historia de la robótica	8
2.1.2. Paradigmas en la robótica	15
2.1.3. Robot Nao	19
2.2. Visión artificial	21
2.2.1. Modelos de color	22
3. Descripción del sistema	26
3.1. Introducción	26
3.2. Análisis del sistema	27
3.2.1. Descripción de las características funcionales	28
3.2.2. Entorno operacional	28
3.2.3. Especificación de casos de uso	29
3.2.4. Especificación de requisitos	38
3.3. Diseño del sistema	49
3.3.1. Arquitectura del sistema	49

3.3.2.	Descripción general del sistema	51
3.3.3.	Descripción de componentes	55
4.	Experimentación	72
4.1.	Experimento 1	73
4.1.1.	Descripción del experimento	73
4.1.2.	Resultados obtenidos	75
4.1.3.	Datos extraídos	75
4.2.	Experimento 2	76
4.2.1.	Descripción del experimento	76
4.2.2.	Resultados obtenidos	77
4.2.3.	Datos extraídos	77
4.3.	Experimento 3	78
4.3.1.	Descripción del experimento	78
4.3.2.	Resultados obtenidos	79
4.3.3.	Datos extraídos	79
4.4.	Experimento 4	80
4.4.1.	Descripción del experimento	80
4.4.2.	Resultados obtenidos	82
4.4.3.	Datos extraídos	82
4.5.	Experimento 5	83
4.5.1.	Descripción del experimento	83
4.5.2.	Resultados obtenidos	84
4.5.3.	Datos extraídos	84
4.6.	Análisis de la experimentación	85
5.	Gestión del trabajo	88
5.1.	Descripción de las fases del trabajo	88
5.2.	Planificación	89
5.3.	Presupuesto	92
5.3.1.	Recursos humanos	92

5.3.2. Recursos físicos	92
5.3.3. Costes indirectos	93
5.3.4. Beneficios y riesgos	93
5.3.5. Cálculo de coste total	94
6. Conclusiones	95
6.1. Conclusiones generales	95
6.2. Conclusiones referentes a los objetivos	96
6.3. Problemas encontrados	98
6.4. Trabajos futuros	99
 A. Manual de instalación	 101
A.1. Instalación del sistema	101
A.2. Instalación de OpenCv	103
A.3. Instalación de Choregraphe	108
 B. Manual de usuario	 109
B.1. Antes de comenzar...	109
B.2. Comience a usar el sistema	109
B.3. Introducción del objeto a buscar	110
B.4. Comienza la búsqueda	111
 C. Robot NAO	 112
C.1. Lista de motores NAO	112
C.2. Posición base o inicial	113

Índice de figuras

2.1. <i>Gallo de Estrasburgo.</i>	8
2.2. <i>Armadura mecanizada de Leonardo da Vinci.</i>	9
2.3. <i>Electro.</i>	10
2.4. <i>Unimate de General Motors.</i>	11
2.5. Robot <i>SHAKY</i>	12
2.6. Robot <i>PUMA</i>	12
2.7. Robot <i>Dante</i>	13
2.8. Robot <i>Dante II</i>	13
2.9. Robot <i>Asimo</i>	14
2.10. UAV Global Hawk.	14
2.11. Rover <i>Spirit</i>	14
2.12. Paradigma deliberativo.	16
2.13. Robot <i>SHAKY</i>	17
2.14. Paradigma reactivo.	18
2.15. Paradigma híbrido.	19
2.16. Robot NAO.	20
2.17. Etapas necesarias para el procesamiento de imágenes.	22
2.18. Modelo de color RGB	23
2.19. Modelo de color HSV	24
2.20. Valores de la luminosidad del formato HSV.	25
3.1. Diagrama de casos de uso.	31
3.2. Modelo de arquitectura Top-Down.	50

3.3. Diagrama del funcionamiento del sistema.	51
3.4. Procesamiento de imagenes para la detección de ojetos.	52
3.5. Centrado de la cámara en una figura.	54
3.6. Posición necesaria para el proceso de recogida.	54
3.7. Diagrama de componentes.	56
3.8. Etapas de aproximación a la mesa.	56
3.9. Etapas de aproximación al objeto.	57
3.10. Etapas de aproximación al objeto.	59
3.11. Punto de referencia para centrar la visión.	59
3.12. Teorema del seno.	61
3.13. Hardware cámara.	62
3.14. Aplicación del teorema del seno.	63
3.15. Aplicación del teorema del seno.	64
3.16. Movimientos de recogida.	66
3.17. Atributos de color de los objetos.	68
4.1. Objetos recogidos en el experimento 1.	74
4.2. Representación experimento 1.	74
4.3. Cono amarillo usado como figura a recoger.	76
4.4. Representación experimento 2.	77
4.5. Representación experimento 3.	79
4.6. Representación experimento 4.	81
4.7. Captación figuras experimento 5.	83
4.8. Representación experimento 5.	84
5.1. Diagrama de Gantt con las horas estimadas.	91
5.2. Diagrama de Gantt con las horas reales.	91
C.1. Motores del robot NAO.	112
C.2. Posición inicial del robot NAO.	114

Índice de tablas

3.1. Caso de uso 001	32
3.2. Caso de uso 002	33
3.3. Caso de uso 003	33
3.4. Caso de uso 004	34
3.5. Caso de uso 005	34
3.6. Caso de uso 006	35
3.7. Caso de uso 007	35
3.8. Caso de uso 008	36
3.9. Caso de uso 009	36
3.10. Caso de uso 010	37
3.11. Caso de uso 011	37
3.12. Requisito funcional 001	40
3.13. Requisito funcional 002	40
3.14. Requisito funcional 003	41
3.15. Requisito funcional 004	41
3.16. Requisito funcional 005	41
3.17. Requisito funcional 006	42
3.18. Requisito funcional 007	42
3.19. Requisito funcional 008	42
3.20. Requisito funcional 009	43
3.21. Requisito funcional 010	43
3.22. Requisito funcional 011	43
3.23. Requisito funcional 012	44

3.24. Requisito funcional 013	44
3.25. Requisito funcional 014	44
3.26. Requisito funcional 015	45
3.27. Requisito no funcional 001	45
3.28. Requisito no funcional 002	45
3.29. Requisito no funcional 003	46
3.30. Requisito no funcional 004	46
3.31. Requisito no funcional 005	46
3.32. Requisito no funcional 006	46
3.33. Requisito no funcional 007	47
3.34. Requisito no funcional 008	47
3.35. Requisito no funcional 009	47
3.36. Requisito no funcional 010	48
3.37. Requisito no funcional 011	48
5.1. Definición de tiempos y tareas del procesos de desarrollo del software.	90
5.2. Costes salariales personal.	92
5.3. Costes físicos.	93
5.4. Costes indirectos.	93
5.5. Beneficios y riesgos del trabajo.	93
5.6. Coste total del trabajo.	94
C.1. Posición inicial del robot NAO.	113

Capítulo 1

Introducción

1.1. Descripción del problema

Este trabajo versa sobre el desarrollo de tareas en el campo de la robótica. La robótica se puede definir como el uso de una entidad mecánica denominada robot mediante la cual realizar tareas de diferente complejidad y de forma automática e independiente. No existe una definición clara de que máquinas pueden ser consideradas robots, pero hay una serie de tareas, de las que una máquina debe realizar algunas o todas ellas, para que pueda ser considerado un robot. Estas tareas son, moverse, mover articulaciones mecánicas, sentir y manipular el entorno o mostrar comportamientos que imiten a los humanos u otros animales. La definición de la robótica realizada por la RAE [1] es “La técnica que aplica la informática al diseño y empleo de aparatos que, en sustitución de personas, realizan operaciones o trabajos, por lo general en instalaciones industriales.” Otra definición ofrecida por Robot Institute of America [2] “Es un dispositivo reprogramable y multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especializados a través de movimientos programados”

A partir de estas definiciones se puede resumir diciendo que la robótica pretende mecanizar tareas manuales por medio de elementos mecánicos. Para realizar estas tareas es necesario combinar diferentes mecanismos y motores con el fin de

dar libertad. Con ello se pretende ahorrar, e incluso disminuir, el esfuerzo humano utilizado en diversas situaciones, tanto en el ámbito cotidiano como en el ámbito laboral.

Actualmente, la robótica combina diferentes áreas tales como la mecánica, la informática, la electrónica y la inteligencia artificial, con el fin de resolver problemas más complejos. Debido a esto los robots modernos son capaces de realizar tareas remotamente en lugares de difícil acceso, con un alto riesgo o con condiciones perjudiciales para el ser humano.

Cabe destacar que tareas que pueden resultar sencillas para el ser humano como pueden ser: buscar un objeto, detectar un obstáculo en el camino, seguir un objeto, etc. Son difíciles de resolver desde la perspectiva de un robot autónomo. Esto se debe a que la forma de procesar la información en un ordenador es diferente a la forma que procesa la información el cerebro humano. Esta diferencia provoca que tareas sencillas de resolver para un humano, requieran un gran desarrollo para su automatización. Bien es cierto que una vez implementada correctamente la tarea, la realización de la misma es automática para un robot y requiere menos esfuerzo que si la realizara una persona.

La creación de funcionalidades básicas para robots es una tarea de gran importancia. Esto se debe a que mediante la agrupación de estas funcionalidades se puede dar lugar a comportamientos más complejos con los que se puedan resolver problemas de mayor dificultad. En este caso trataremos de aportar al robot NAO una serie de funcionalidades que le permitan encontrar y recoger un objeto de forma autónoma. Este comportamiento, como ya se ha dicho, pretende aportar una base para la realización de trabajos más avanzados dando una serie de funcionalidades útiles para diversas tareas.

1.2. Motivación

La creación de funcionalidades y actitudes básicas de detección de objetos, acercamiento y recogida sobre el robot NAO permite aportar facilidades para

trabajos localización y recogida de objetos, como puede ser, localizar una pelota para recogerla, mostrar su posición o seguir su trayectoria.

Dentro de este ámbito las acciones que se pueden realizar interactuando con objetos del entorno son ilimitadas. Por ello, como primer paso para lograr realizar estas tareas es necesario partir de una funcionalidad que sirva como una base sólida con el fin de desarrollar sobre ella un sistema con actividades más complejas. De esta necesidad surge este trabajo, como parte común en todas las acciones que impliquen interactuar con objetos es necesario realizar las siguientes tareas: localización de un objeto en un entorno, posicionar ese objeto en el espacio tomando como referencia el robot y realizar un acercamiento que permita situar al robot en una posición con el fin de interactuar con el propio objeto. Desarrollar estas primitivas de forma correcta significa dar facilidades para realizar un sistema que, entre otros ejemplos, tenga por objetivo recoger y colocar objetos de forma autónoma o dar una patada a una pelota.

1.3. Objetivos del trabajo

El objetivo principal de este trabajo consiste en crear un sistema capaz de localizar y recoger un objeto situado sobre una mesa. El proceso se tiene que realizar de forma autónoma mediante la utilización de un robot humanoide Nao y sin conocer la posición del objeto antes de comenzar la ejecución. La extracción de información del entorno se realizará mediante visión artificial a través de las cámaras disponibles en el robot. Así mismo se tiene que implementar el conjunto de movimientos que realizará el robot Nao para una recogida correcta del objeto.

Partiendo de lo descrito en el apartado anterior se han definido un conjunto de objetivos que sirvan de guía para la realización del trabajo. Estos objetivos ayudaran a decidir los pasos que se tomarán en el desarrollo del sistema hasta llegar al resultado final. A continuación se realiza una descripción detallada de los objetivos:

1. Estudio previo

Estudiar el framework NAOqi y familiarizarse con sus características para aprovechar sus capacidades. Así mismo se realizará un estudio de las diferentes técnicas de visión artificial disponibles para ser capaz de realizar un desarrollo acorde con las necesidades del trabajo mediante librerías ya existentes o realizando una implementación de funciones específicas.

2. Detección de objetos

Se deberá desarrollar un sistema que procese las imágenes obtenidas del entorno con la finalidad de localizar un determinado objeto dentro del mismo. La detección se deberá realizar de forma efectiva e inequívoca debido a la importancia de este proceso dentro del trabajo. Así mismo, para que el funcionamiento sea el adecuado, se realizará un tratamiento con el cual se desechará todo el ruido posible contenido en la imagen.

3. Extracción de información del entorno

Se deberá desarrollar un proceso que extraiga toda la información del entorno necesaria con el fin de cumplir el objetivo principal. Para la extracción de esta información se utilizarán los medios necesarios mediante el uso, tanto de los elementos aportados por el robot NAO con el framework NAOqi, como cualquier otro elemento externo.

4. Implementación de primitivas motrices para el robot NAO

Se llevará a cabo una implementación de los comportamientos necesarios sobre el robot NAO usando los módulos disponibles en el framework NAOqi.

5. Agrupación de todos los componentes

Se realizará una combinación de las funcionalidades realizadas anteriormente con el fin de resolver el objetivo general. Se prestará una atención especial para realizar un acoplamiento adecuado de los distintos módulos y sistemas y garantizar así un correcto funcionamiento del sistema.

6. Evaluación del sistema

Se realizará una serie de experimentos para asegurar que el funcionamiento sea el adecuado y que los objetivos se han cumplido. Así mismo estos experimentos servirán para determinar los límites del sistema.

7. Desarrollo de la documentación

Se documentará el trabajo realizado. En esta documentación se incluirá la realización de cada una de las partes anteriormente descritas así como el material de ayuda y de interés relacionado con el trabajo realizado.

1.4. Estructura del documento

El contenido de este documento se describe brevemente a continuación:

- En el primer capítulo se hace una introducción al trabajo realizado mostrando la motivación y los objetivos del mismo.
- En el segundo capítulo se expone el estado del arte. En este capítulo se realiza una descripción detallada de la áreas de conocimiento relacionadas con este trabajo. Primero se mostrará el contexto histórico de la robótica para después explicar los diferentes paradigmas de control utilizados en robótica. A continuación se realiza una descripción detallada de algunos conceptos de visión artificial, haciendo inciso en los modelos de color y en los pasos para realizar una percepción de objetos.
- En el tercer capítulo se realiza una descripción del sistema que ha sido desarrollado en este trabajo. En primer lugar se realizará el análisis del sistema que se va a desarrollar a través de los casos de uso y de los requisitos. La segunda parte se corresponde con el diseño del sistema y se especificará los componentes que lo forman.
- En el cuatro capítulo se presenta la experimentación realizada sobre el sistema desarrollado. Se realizarán un conjunto de experimento con los que se pretende comprobar el grado de éxito del trabajo.

- En el quinto capítulo se presenta la planificación del trabajo. Aquí se muestra la distribución temporal en las distintas fases de desarrollo del trabajo así como una estimación del presupuesto necesario para llevarlo a cabo.
- En capítulo seis se presenta las conclusiones extraídas de la realización del trabajo de fin de grado. Así mismo se mostrarán los problemas obtenidos y los posibles trabajos futuros a realizar sobre el sistema.
- Por último se adjuntarán los anexos en los que se amplía información del trabajo. Aquí se explicaran los pasos para una correcta instalación de los elementos necesarios para ejecutar el sistema y un manual de usuario.

Capítulo 2

Estado de la Cuestión

A continuación se presenta una breve explicación del estado de la cuestión en el que se enmarca este trabajo. En este apartado se hablará sobre la historia de la robótica, haciendo hincapié en los paradigmas de la robótica, y se realizará un breve estudio sobre la visión artificial.

2.1. Robótica

En 1920 el escritor checo Karel Capek escribió una obra de teatro llamada *Rossum's universal robots* [3]. En esta obra, el gerente de una fábrica construía unas maquinas que se ponían al servicio del hombre realizando todas las tareas mientras los humanos se dedicaban al ocio permanente. En esta obra se utiliza por primera vez el término robot derivada del checo *robota*, que significa «siervo, esclavo, trabajador forzado».

La robótica es un concepto de dominio público. Hoy en día la mayor parte de la gente tiene una idea de lo que es la robótica, conoce sus posibles aplicaciones y el potencial que tiene; sin embargo, desconocen del origen de la misma y de la multitud de aplicaciones prácticas de la robotica. La robótica como hoy en día la conocemos, tiene sus orígenes hace cientos de años. Los robots eran conocidos con el nombre de autómatas, y la robótica no era reconocida como ciencia [4]. A

continuación se realizará un breve resumen sobre la historia de la robótica.

2.1.1. Historia de la robótica

El concepto de máquinas automatizadas se remonta a la antigüedad, donde existen muchos relatos de seres mecánicos similares a humanos, los cuales se comportan igual que nosotros y su trabajo era obedecer órdenes. Estas máquinas eran llamadas autómatas y han ido surgiendo en diversas culturas hasta el día de hoy. Durante el transcurso de los años, a medida que mejoraba la tecnología, se han desarrollado máquinas especializadas para tareas específicas. Sin embargo ninguna de estas máquinas tenía la versatilidad de un ser humano.

Existen un gran número de apariciones de automatas a lo largo de los años y su origen se remonta a la antigua Grecia. A continuación se muestran algunos autómatas desarrollados, capaces de imitar ciertos tipos de comportamientos o de realizar tareas complejas:

- Un ejemplo es el Gallo de Estrasburgo que funcionó desde 1352 hasta 1789 (Figura 2.1). Este es el autómata más antiguo que se conserva en la actualidad, formaba parte del reloj de la catedral de Estrasburgo y al dar las horas movía el pico y las alas.

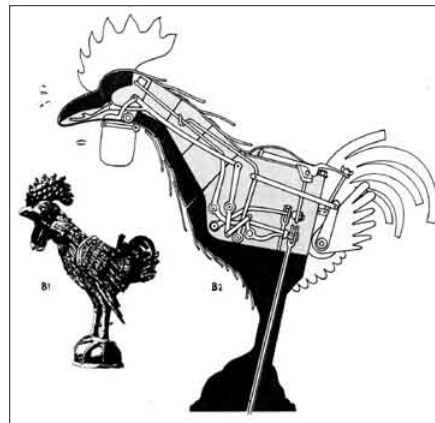


Figura 2.1: *Gallo de Estrasburgo*.

- Leonardo da Vinci diseñó una maquina con el aspecto de un caballero con armadura capaz de saludar en las recepciones reales (Figura 2.2). Este autómatas se considera el primero con forma humana aunque no se tiene constancia de que lo construyera realmente.



Figura 2.2: *Armadura mecanizada de Leonardo da Vinci.*

- Pierre Jaquet-Droz construyó tres autómatas que se presentaron al público por primera vez en 1774. Algunos consideran estos autómatas como los predecesores de los robots modernos:
 - **La pianista**, una autómatas de dos mil quinientas piezas capaz de interpretar de manera real una partitura al órgano con sus propios dedos.
 - **El dibujante**, de dos mil piezas, un niño capaz de realizar hasta cuatro dibujos diferentes desde el esbozo en lápiz hasta los retoques finales.
 - **El escritor**, un autómatas de seis mil piezas capaz de escribir a pluma diferentes textos en inglés y francés siguiendo con su mirada lo que escribe.
- Nikola Tesla presentó en la prestigiosa Feria de la Electricidad del Madison Square Garden, en 1898, el primer prototipo de barco tripulado completamente teledirigido. De hecho, para Tesla, aquella patente era el primer paso para alcanzar un futuro repleto de máquinas que no solo se pudieran controlar a distancia, sino que desarrollaran su propia inteligencia y autonomía,

capaces de ejecutar todas aquellas penosas, duras, peligrosas o simplemente tediosas tareas que el hombre dejaría de realizar para siempre [5].

- En 1939 la empresa estadounidense Westinghouse presentó en la feria mundial celebrada en Nueva York a *Elektro*. *Elektro* era un humanoide que medía 2 metros y 10 centímetros y pesaba 120 kilos, aproximadamente. Era capaz de realizar 26 movimientos diferentes, así como fumar y caminar. También tenía los sentidos desarrollados: podía ver gracias a unas células fotoeléctricas insertadas en sus ojos, y hablar, fruto de un aparato reproductor incorporado con un disco en el que había grabadas 700 palabras. En la figura 2.3 se puede ver una imagen de *Elektro* en la feria mundial celebrada en Nueva York. Esta presentación supuso un importante paso hacia el futuro de la robótica.



Figura 2.3: *Electro*.

Después de la segunda guerra mundial se inician programas de investigación para desarrollar manipuladores mecánicos para elementos radiactivos. Estos manipuladores consistían en brazos robóticos diseñados para reproducir el movimiento de brazos y manos de un operador. En la segunda mitad del siglo XX los avances en electrónica e informática dieron lugar a una verdadera explosión de la robótica.

En este periodo de tiempo nacieron una gran cantidad de grupos de investigación y empresas centrados en el campo de la robótica. Se puede considerar este punto como el inicio de la era de la Robótica tal como la conocemos [6].

El primer ejemplo de este gran avance se da en 1961, cuando la empresa de General Motors instaló en su cadena de montaje el primer robot industrial, el *Unimate*. Esta máquina, que puede verse en la imagen 2.4, realizaba trabajos que resultaban peligrosos para los trabajadores como, transportar piezas fundidas y soldar estas piezas al chasis. El *Unimate* fue creado en los años 50 por George Devol [7], quien fundó la primera empresa fabricantes de robots, *Unimation*.

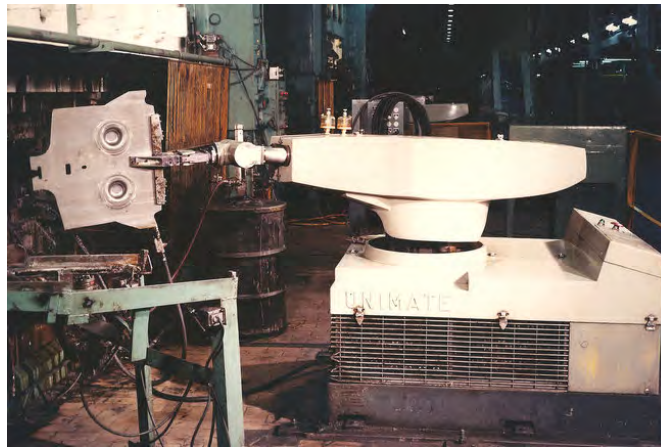


Figura 2.4: *Unimate de General Motors.*

Durante la década de los 60 se hace necesaria una mayor flexibilidad en relación con los avances anteriores. Para ello se comienza a introducir realimentación sensorial en los nuevos sistemas desarrollados. Con esto se pretendía que el robot pudiera reaccionar dependiendo de la naturaleza del entorno con el fin de ofrecer un mayor grado de autonomía.

Siguiendo estas directrices se construyó en 1970 el primer robot capaz de percibir su entorno y planificar sus acciones en base al mismo. Este robot se denominó *SHAKY* (Figura 2.5). Fue construido por el centro de inteligencia artificial del Stanford Research Institute y estaba dotado de un escáner, cámara de visión y sensores táctiles [8]. El robot estaba controlado por un programa externo llamado *STRIPS*, cuyo objetivo era realizar una secuencia de acciones para llegar a un

estado final del entorno. En seguida quedó de manifiesto la dificultad de diseñar programas de control de un robot autónomo y reflejándose que, cuanto más real es el entorno, mayor dificultad se encontraba para completar con éxito la tarea.

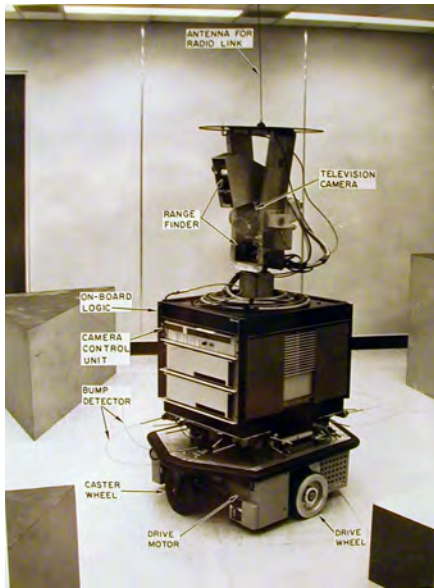


Figura 2.5: Robot *SHAKY*.



Figura 2.6: Robot *PUMA*.

En 1975, Will y Grossman, en IBM, desarrollaron un manipulador controlado por computador con sensores de contacto y fuerza para montajes mecánicos. Este mismo año, el ingeniero mecánico estadounidense Victor Scheinman, cuando estudiaba la carrera en la Universidad de Stanford, California, desarrolló un manipulador polivalente realmente flexible conocido como Brazo Manipulador Universal Programable (PUMA, siglas en inglés). El PUMA (Figura 2.6) era capaz de mover un objeto y colocarlo en cualquier orientación en un lugar deseado que estuviera a su alcance. El concepto básico multiarticulado del PUMA es la base de la mayoría de los robots actuales [9].

En 1992 la Universidad de Carnegie Mellon diseñó y construyó un robot llamado *Dante*. Este robot era capaz de subir y bajar pendientes pronunciadas y estaba diseñado para explorar cráteres volcánicos activos. En términos de la robótica los objetivos de *Dante* eran demostrar una verdadera misión de exploración, locomoción de terrenos difíciles y supervivencia del medio ambiente. Sin

embargo fracasó en su labor debido a un cable de fibra óptica defectuoso [10]. En 1994 se diseñó una nueva versión, el *Dante II* [11], que consiguió recoger gases del volcán Spurr en Alaska. Tanto el robot Dante como el Dante II pueden verse en las figuras 2.7 y 2.8 respectivamente.



Figura 2.7: Robot *Dante*.



Figura 2.8: Robot *Dante II*.

A partir del 2000 hay una rápida explosión en el campo de la robótica gracias a los avances anteriores. Algunos de los hitos de más importancia:

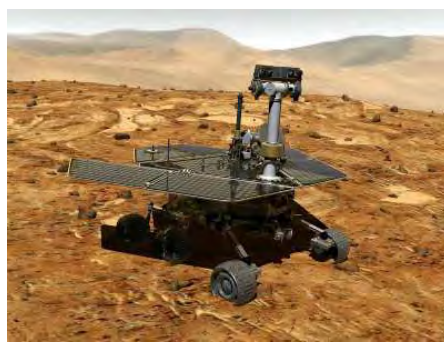
- La compañía japonesa de Honda presenta en el 2000 el robot *ASIMO* (figura 2.9), un robot humanoide con una altura de 1,88m capaz de andar simulando el caminar humano [12]. En la última versión del 2007 el robot cuenta con varias aplicaciones procedentes de la inteligencia artificial: puede identificar y coger objetos, subir y bajar escaleras, entender y dar respuesta a órdenes orales e incluso reconocer las caras de algunas personas.
- La versatilidad proporcionada por los robots autónomos provoca el desarrollo de los UAV (Unmanned Aerial Vehicle) con el objetivo de realizar misiones de reconocimiento y vigilancia en zonas de alto peligro. Actualmente existe

Figura 2.9: Robot *Asimo*.

Figura 2.10: UAV Global Hawk.

un gran número de UAV (y un creciente número de UCAV (Unmanned Combat Air Vehicle)) de los que podemos destacar el Global Hawk (Figura 2.11), Elbit Hermes y el EADS Barracuda entre otros.

- Actualmente la robótica está entrando también en el ámbito doméstico. En 2002 se lanzó al mercado el *Roomba*, un robot aspiradora capaz de recorrer todas las estancias de la casa y reaccionar a los cambios que se producen en su entorno.
- En 2003 la robótica ofrece un salto en la exploración de otros planetas al ser enviados por la NASA a la superficie de Marte los rovers autónomos *Spirit*, cuya imagen se observa en la figura 2.11 y *Opportunity*.

Figura 2.11: Rover *Spirit*.

2.1.2. Paradigmas en la robótica

Paradigma es un término de origen griego, "*parádeigma*", que significa modelo, patrón, ejemplo. En un sentido amplio se corresponde con algo que va a servir como modelo o ejemplo a seguir en una situación dada. Una definición más amplia de esta palabra es la dada por Thomas Kuhn en su libro "La estructura de las revoluciones científicas" [13], definió paradigma como los "logros científicos que generan modelos que, durante un período más o menos largo, y de modo más o menos explícito, orientan el desarrollo posterior de las investigaciones exclusivamente en la búsqueda de soluciones para los problemas planteados por estas".

El paradigma es un principio, una teoría o un conocimiento originado de la investigación en un campo científico. Una referencia inicial que servirá como modelo para futuras investigaciones.

En la robótica nos encontramos con 3 paradigmas, el deliberativo, el reactivo y el híbrido. A continuación se mostrarán las características de cada uno de ellos.

2.1.2.1. Paradigma deliberativo

El paradigma deliberativo o jerárquico es el más antiguo. Históricamente tuvo su mayor influencia en las décadas de los 60 y 70. Este paradigma se basa en gran medida en una aproximación antropomórfica, bajo la cual los robots se desarrollan tomando como inspiración la manera en la que los seres humanos resuelven los problemas. Esto supone que este paradigma tiene mucho de simbólico, y se da mucha importancia a las tareas de planificación, más si cabe teniendo en cuenta que en la época en la que surgió, la aproximación simbólica a la inteligencia artificial se encontraba en su momento de mayor auge.

Este paradigma se denomina también paradigma Percepción, Planificación, Acción ó PPA, por los nombres de las tres operaciones básicas que lo componen [14].

La denominación de jerárquico se debe a que las tres operaciones que aca-

bamos de mencionar se llevan a cabo siempre secuencialmente, en el orden en que las hemos nombrado, y porque cada operación toma como entrada las salidas de la operación anterior.

La **percepción** es clave en todo robot autónomo, ya que para poder influir sobre el entorno, es necesario que exista una manera de percibir el estado actual del mismo.

En la fase de **planificación** el robot decide qué acciones llevar a cabo tomando como base las percepciones recién adquiridas y sus propios objetivos. Para poder efectuar la planificación, las percepciones deberán haber sido preprocesadas y transformadas en una abstracción del entorno.

En la fase de **acción** se llevan a cabo las acciones de alto nivel que se seleccionaron en la fase de planificación, traduciéndolas en movimientos de los actuadores del robot.

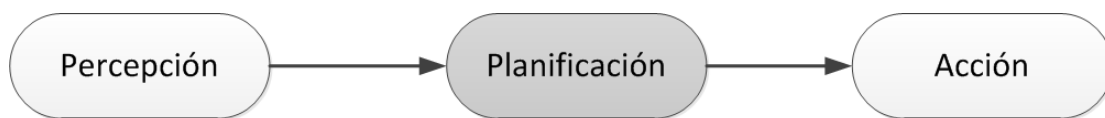
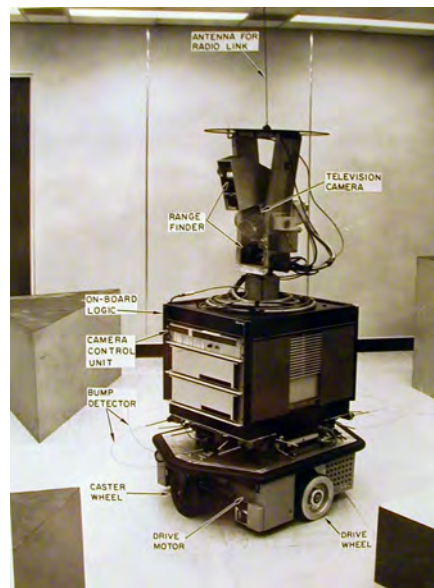


Figura 2.12: Paradigma deliberativo.

De acuerdo con el paradigma deliberativo primero se ha de percibir el entorno, después planificar la acción a realizar y finalmente ejecutar dicha acción. De esta forma cada vez que se pretende realizar una tarea mediante el paradigma deliberativo ha de completarse el ciclo completo pasando por las tres operaciones de manera independiente (figura 2.12). Debido a esto en el paradigma deliberativo es necesaria la realización de un modelo interno del mundo real que sea totalmente completo y correcto. La realización de este modelo interno por lo tanto es la parte más determinante de las operaciones ejecutadas. Para la realización del mismo es necesario llevar a cabo un procesamiento inicial de todos los datos recogido por los sensores provocando que la fase de acción no pueda realizarse hasta haber realizado el modelo del entorno. Debido a ello este modelo de desarrollo de inteligencia artificial es inadecuado en la realización de tareas en tiempo real. Como punto a favor de la realización de un modelo interno cabe destacar que, al realizar un

Figura 2.13: Robot *SHAKY*.

procesamiento de todos los datos recogidos por los sensores la presencia de ruido en alguno de los mismos no resulta ser determinante a la hora de seleccionar una acción a tomar.

En la época en la que se desarrolló este paradigma se consideraba que los grandes problemas que planteaban las técnicas de inteligencia artificial eran los relativos a la demostración de teoremas matemáticos, la comprensión del lenguaje natural y otras tareas formales. Estos fueron los temas fundamentales que se trataron en la conferencia de Dartmouth en el año 56. Los primeros trabajos surgidos de aquí fueron los que dieron origen a la Inteligencia Artificial tal y como la conocemos, y centraron las tareas a las que se dedicarían los investigadores en los años siguientes.

Posteriormente los investigadores comenzaron a darse cuenta de que las tareas realmente complejas son las relacionadas con la percepción. A principios de los años 60 se consideraba que problemas como el de la visión artificial se solucionarían definitivamente en un plazo relativamente corto. Sin embargo la práctica demostró que este tipo de tareas presentan muchas más dificultades que las que tienen que ver con el razonamiento formal. El ejemplo más claro de la utilización

del paradigma deliberativo en el campo de la inteligencia artificial es el programa *STRIPS*, utilizado en el robot *SHAKY* (Figura 2.13).

2.1.2.2. Paradigma reactivo

El paradigma reactivo fue la respuesta a los problemas que introducía el paradigma deliberativo. Este paradigma, que surgió en el año 1986 [15] y estuvo vigente hasta principios de los noventa, está inspirado en las ciencias cognitivas y en la biología, y como su nombre indica su manera de operar obvia la planificación y se basa en correspondencias directas entre percepción y acción [16], como se observa en el diagrama 2.14. Mientras que el paradigma deliberativo asume que la acción a realizar es el resultado de una planificación, el paradigma reactivo obtiene que la acción se vea influenciada directamente por la entrada de un sensor realizando un mínimo de procesamiento.

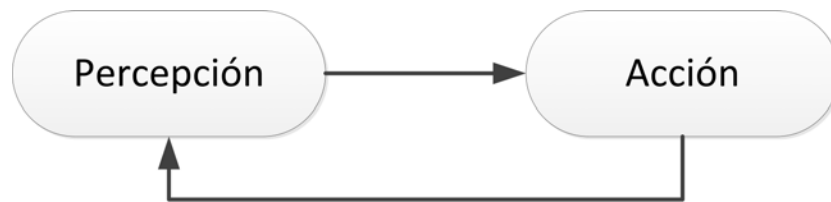


Figura 2.14: Paradigma reactivo.

El paradigma PA presenta unos requerimientos mucho menores para las tareas de percepción, ya que al no haber planificación no resulta necesario abstraer lo percibido transformándolo en símbolos, con lo que se evita la explosión combinatoria. En lugar de ello, la información percibida es enviada al módulo de acción con un mínimo preprocesamiento. El módulo de acción trabaja en base a estas entradas subsimbólicas y a partir de ellas genera un comportamiento. Este modelo de computación es idóneo para abordar problemas que necesiten acciones en tiempo real para su realización.

A pesar de su mayor sencillez, o precisamente debido a ella, los robots contruidos empleando este paradigma sólo pueden emplearse para llevar a cabo tareas moderadamente sencillas, y que no implican razonamientos complejos.

2.1.2.3. Paradigma híbrido

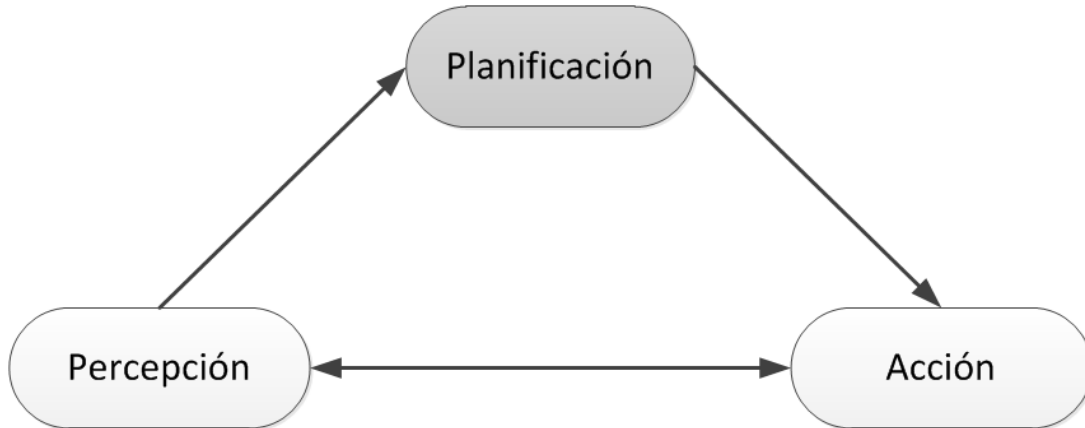


Figura 2.15: Paradigma híbrido.

El paradigma híbrido surgió en los años 90 [17] para evitar las deficiencias del paradigma deliberativo y reactivo. Este modelo se basa en la acción a través de la información de los sensores de manera directa, pero a la vez incorpora un nivel de razonamiento a través de un modelo interno.

La idea de este paradigma es llegar a un compromiso entre las tareas deliberativas y las reactivas. En este caso el robot está compuesto, por un lado, de una cadena deliberativa típica (PPA) y por otro lado de un modelo reactivo (PA) (figura 2.15).

La cadena deliberativa se encarga de llevar a cabo la planificación de alto nivel, como en el caso del modelo deliberativo tradicional. La diferencia es que se reduce el énfasis en la abstracción de las percepciones.

2.1.3. Robot Nao

El robot Nao es un robot humanoide programable y autónomo, desarrollado por Aldebaran Robotics [18], una compañía de robótica francesa con sede en París. Nao fue creado con el objetivo de ofrecer una plataforma hardware y software que permitiera un avance en las investigaciones en un ámbito doméstico.



Figura 2.16: Robot NAO.

Este robot mide 58cm de altura y pesa 4,8 Kg. Está provisto de una batería de ion de litio que le da una autonomía de 90 minutos. Junto a esto cuenta con 25 grados de libertad que le dan la posibilidad de ser completamente autónomo. Cuenta con 2 manos prensiles y articulación en las muñecas. Su caminado omnidireccional es de utilidad para aplicaciones que impliquen mapeo y autonomía. En la figura 2.16 se puede observar una imagen del robot NAO en su versión 4.0. Las características Hardware y Software del robot son las siguientes:

- CPU ATOM Z530 1.6 GHz.
- Memoria Flash de 256 MB SDRAM / 2 GB.
- Sensor de Inercia con Giroscopio de dos ejes y Acelerómetro de 3 ejes.
- 1x Puerto Ethernet RJ45 – 10/100/1000 base T y Wi-Fi IEEE 802.11b/g.
- 2x Cámaras de video (960p@30fps), mejor sensibilidad en VGA. Visión horizontal de 239, visión vertical de 68. Resolución de alta definición.

- Capacidad de procesamiento de visión.
- Reconocimiento de objetos.
- Detección y reconocimiento de rostros.
- Dos altavoces y síntesis vocal multi-idioma (Español e Inglés precargados).
- Cuatro micrófonos y reconocimiento de voz multi-idioma (Español e Inglés precargados).
- Soporta múltiples lenguajes de programación.
- Cuenta con software especial de programación y simulación.

Para nuestro trabajo resulta necesario el uso de las cámaras, las cuales servirán para captar el entorno cercano del robot. Además necesitaremos acceder y modificar los valores de los motores de las articulaciones con el fin de realizar diversos movimientos como, andar, mirar a los lados y recoger un objeto.

2.2. Visión artificial

La visión artificial o visión por computador es un subcampo de la inteligencia artificial, cuyo propósito es extraer la información de una imagen o conjunto de imágenes con el fin de que el ordenador pueda “obtener” el entorno o determinadas características del mismo [19].

Aunque este cometido parece sencillo a simple vista es necesario realizar varias tareas para realizar un reconocimiento de objetos efectivo partiendo de cero. Existen una serie de etapas, que se muestran en la figura 2.17, por las que debe pasar un sistema de visión artificial con el fin de realizar tareas de reconocimientos y localización de objetos, estimación de forma, etc. Dependiendo de las necesidades del sistema desarrollado no siempre es necesario pasar por todas estas fases que se describen a continuación:

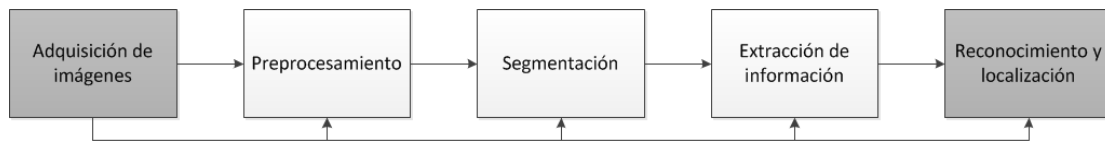


Figura 2.17: Etapas necesarias para el procesamiento de imágenes.

- **Adquisición de la imagen:** Plasma el mundo real en una imagen digital.
- **Preprocesamiento:** Realiza operaciones encaminadas a preparar la imagen para usarla en posteriores etapas.
- **Segmentación:** Determina la región de la imágenes con atributos de interés en la imagen.
- **Extracción de información:** Obtiene una representación matemática de los atributos de la imagen o del segmento de la imagen.
- **Reconocimiento y localización:** Realiza las operaciones sobre los atributos de la imagen para dar como resultado una salida del proceso.

A esto se unen dificultades debido a la naturaleza cambiante del entorno como puede ser, diferencias de luz según el momento del día, la distancia existente al objeto en la imagen, ruidos que puedan estar presente en la imagen y que puedan generar falsos positivos, etc. Debido a esto en el trabajo nos centraremos únicamente en el reconocimiento de objetos por medio de las características de color.

2.2.1. Modelos de color

El modelo de color es una abstracción matemática que permite representar los colores de forma numérica [20]. Existen un gran número de modelos de color, pero para la realización de este trabajo nos centraremos en dos modelos, el modelo RGB y el modelo HSV. El modelo RGB se ha estudiado debido a que es un modelo de color con mucha aceptación y de la que se dispone gran cantidad de ayuda

para el desarrollo del software, siendo este modelo el utilizado para guardar la información de la imagen por el framework de NAOqi. Por parte del modelo de color HSV se ha estudiado debido a las facilidades que ofrece la representación matemática de los colores para realizar la computación de los mismos.

2.2.1.1. Modelo RGB

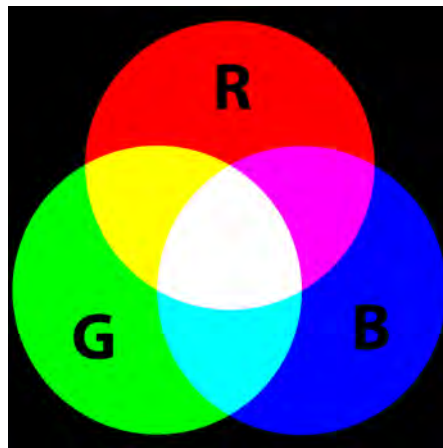


Figura 2.18: Modelo de color RGB

El modelo de color RGB (en inglés Red, Green, Blue) se basa en la síntesis aditiva, según la cual es posible representar cualquier color mediante la combinación de los tres colores primarios. Este modelo se representa matemáticamente como una tupla de 3 valores, de 0 a 255, que simbolizan la cantidad de cada color primario, rojo, verde y azul, como se puede observar en la figura 2.18. De la mezcla de estos primarios se puede obtener el resto de colores, encontrándose en los extremos el blanco y el negro, es decir un color cuyo valor RGB es 0,0,0 es el color negro, y el color cuyo valor es 255, 255, 255 es el color blanco

El modelo RGB es el modelo más básico existente y es usado por un gran número de aplicaciones debido a su sencillez. Por esto, alguno de los otros modelos de color existentes se basan en modificaciones o transformaciones de este modelo.

2.2.1.2. Modelo HSV

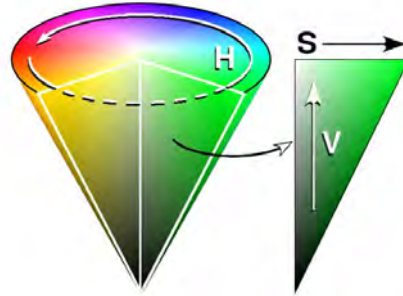


Figura 2.19: Modelo de color HSV

El modelo de color HSV (en inglés Hue, Saturation, Value) se trata de una transformación no lineal del espacio de color RGB. La representación puede verse en la figura 2.19. Este modelo se representa matemáticamente como una tupla de 3 valores:

- **Tono o Matiz:** Se representa como un grado de ángulo cuyos valores posibles van de 0 a 360 siendo 0 el rojo, 60 el amarillo y 120 el verde. Cada valor de esta componente se corresponde únicamente con un color. También puede encontrarse normalizado en otro rango de valores según su utilización.
- **Saturación:** Este valor representa la distancia al eje de brillo negro-blanco. Los valores posibles van del 0 al 100%. Cuanto menor sea la saturación de un color, mayor tonalidad grisácea habrá y más decolorado estará.
- **Valor o Luminosidad:** Representa la altura en el eje blanco-negro. Los valores posibles van del 0 al 100

Este modelo de color resulta muy útil para detectar un determinado color en el entorno. Esto se debe a que se encuentra separado el tono del color del resto de valores facilitando así las comprobaciones.

Por otra parte al tener separado la luminosidad se puede realizar un tratamiento especial para detectar el color con condiciones de luz diferentes. En la

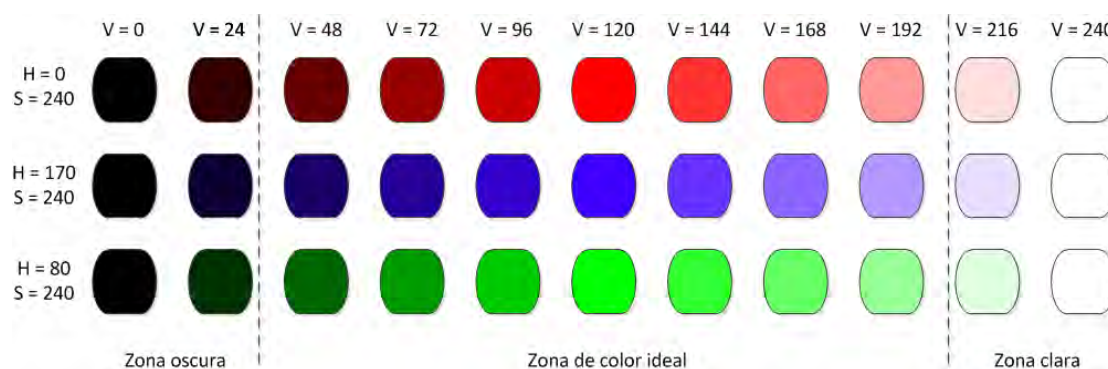


Figura 2.20: Valores de la luminosidad del formato HSV.

figura 2.20 se puede observar cómo, modificando el valor de la luminosidad para una misma tonalidad y saturación los resultados obtenidos son en esencia el mismo color pero con condiciones de luz diferentes. Cabe destacar que la representación de un mismo color en formato RGB modificando la condición de luminosidad modifica sus tres atributos, esto dificulta en gran medida los cálculos necesarios para comprobar si dos colores son parecidos. Además en la imagen se observa una zona con colores demasiado oscuros y otra con colores demasiado claros. Estas zonas servirán para descartar objetos con una luminosidad muy pobre o excesiva, así como sombras y reflejos presentes en las imágenes.

Capítulo 3

Descripción del sistema

En este capítulo se realiza una descripción del sistema y del proceso de desarrollo que se ha seguido para su implementación.

A continuación en el capítulo *3.1. Introducción* y *3.2. Análisis del sistema* se llevará a cabo un análisis del trabajo a desarrollar, el análisis del sistema es una de las tareas más importantes en el ciclo de vida del desarrollo de software, ya que de ellos depende el correcto funcionamiento del nuevo sistema. Para realizar la descripción total del sistema que se pretende desarrollar se identificarán y documentarán todas las partes del mismo. Con ello se obtendrá como resultado final el catálogo de requisitos. En este catálogo de requisitos se sintetizará toda la información recopilada dando las directrices finales de cómo será el sistema generado.

Seguidamente en el capítulo *3.3. Diseño del sistema* se detallará la arquitectura que se utilizará finalmente y se desarrollará la especificación detallada de los componentes del sistema.

3.1. Introducción

En este trabajo se pretende resolver el problema de búsqueda, localización, acercamiento y recogida de un objeto. Para ello se deben realizar distintas etapas con el fin de llegar al objetivo final. Estas etapas son:

- Reconocimiento del entorno: Se deberá tomar imágenes del entorno y procesarlas extrayendo la información con el fin de localizar los objetos que se encuentren en ellas. Para ello se creará un módulo de visión que se encargará de segmentar la imagen localizando las zonas con atributos de color similares. Además se extraerá la información de cada segmento para realizar el procesamiento oportuno de los mismos.
- Localización: El sistema deberá utilizar la información extraída de la imagen para determinar las cualidades de cada objeto. Así como la situación del mismo con respecto al robot si se considera oportuno.
- Acercamiento: Habiendo localizado un objeto es necesario realizar un acercamiento a él con el fin de interactuar con el mismo. Para ello se utilizara la capacidad motora del robot mediante los motores de sus articulaciones con el fin de posicionarlo en una posición con respecto al objeto.
- Interactuación con el objeto: Se realizará un comportamiento del robot que permita interactuar con el objeto recogiénolo de la posición en la que se encuentra. Para ello de nuevo será necesario utilizar los motores de las articulaciones del robot.

3.2. Análisis del sistema

A continuación se describirá el proceso de análisis que se ha realizado para construir el sistema. El objetivo de esta fase es llevar a cabo una especificación profunda y detallada de que debe realizar el sistema y como lo debe hacer. Esto servirá de base durante el posterior diseño de la aplicación.

3.2.1. Descripción de las características funcionales

El sistema que se va a desarrollar tiene como objetivo encontrar y recoger un objeto. Para conseguir este objetivo el sistema debe tener las siguientes características funcionales:

El sistema será capaz de manejar un robot NAO buscando a su alrededor los elementos existentes. Esta búsqueda se realizará procesando las agrupaciones de colores de la imagen recogida por una de las dos cámaras del robot. Para la búsqueda de los objetos el sistema realizará un procesamiento de la imagen en busca de todos los elementos existentes, seguidamente el usuario deberá seleccionar el objeto a recoger. Una vez introducida esta información en el sistema, este guardará los datos del color para su posterior búsqueda.

Al lanzar la ejecución el sistema realizará otra búsqueda buscando el color de la mesa y de la figura que se debe recoger. Con el fin de recoger la figura, en primer lugar el sistema calculará la distancia existente hasta la mesa y se acercará a ella. Cuando se encuentre suficientemente cerca de la mesa se realizará la búsqueda al objeto y se calculará la distancia al mismo. Finalmente antes de hacer efectiva la recogida del objeto se realizará un acercamiento en el que el robot se moverá para colocarse en una posición determinada. Una vez realizado esto el sistema ejecutará un comportamiento para la recogida del objeto finalizando la ejecución.

3.2.2. Entorno operacional

A continuación se detallará el entorno en el que el sistema se ejecutará. Se detallarán los elementos que intervienen, las conexiones necesarias y las características que se deben cumplir para hacer uso del sistema desarrollado.

En primer lugar para necesitaremos, un robot NAO en su versión 4.0 con el framework NAOqi con la versión 1.12.3. Se utilizará un ordenador sobre el que se ejecutará el sistema. Por ultimo para poder utilizar el sistema es necesario realizar una conexión entre el robot y el equipo en el que se realiza la ejecución. Para ello existen dos maneras de realizar la conexión, la primera mediante un cable ethernet conectado a las dos partes, y la segunda se utilizaría un router wifi creando una red local entre el robot y el equipo.

Además de esto se necesitarán diversos objetos para la ejecución completa del sistema. El primero deberá ser un objeto de dimensiones reducidas y preferiblemente de gomaespuma para que pueda ser agarrado por la mano del robot. Junto a esto se necesitará una caja que se usará de apoyo para el objeto. Esta caja actuará de mesa y deberá tener una altura de 31 cm para que la recogida del objeto sea óptima. Para evitar problemas en el reconocimiento de colores se aconseja que la superficie de la mesa sea de color blanca o negra y de textura lisa.

3.2.3. Especificación de casos de uso

A continuación se van a mostrar los casos de uso identificados en el análisis del sistema. En primer lugar se mostrará el conjunto de casos de uso y actores identificados por medio de un diagrama. Más adelante se hará una descripción de cada caso de uso en profundidad definiendo la actividad realiza por el actor.

3.2.3.1. Descripción de los actores

Los actores son entidades externas al sistema que estamos construyendo y que interactúan con el siempre de la misma forma. Por lo tanto, al identificar actores estamos empezando a delimitar el sistema, y a definir su alcance. Los actores que interactuaran con el sistema final serán los siguientes:

- Usuario: Este actor interactuará con el sistema únicamente para introducir el objeto que se buscará una vez iniciado el sistema.

- Robot: Este actor interactuará con el sistema ofreciéndole un mecanismo para localizar y recoger un objeto. Para ello se usaran de todas las funcionalidades necesarias del robot Nao.

3.2.3.2. Descripción de los atributos de los casos de uso

Para la realización de la descripción textual de los distintos casos de uso, se han seleccionado una serie de atributos que los describen adecuadamente. A continuación se realiza una descripción del significado de cada uno de los atributos utilizados para la descripción de los casos de uso.

- Código: Identificación unívoca abreviada del caso de uso, se construye mediante CU seguido de un - y de tres dígitos. Por ejemplo CU-001.
- Nombre: Identificación extendida del caso de uso.
- Actores: Conjunto de entidades que interactúan con el caso de uso. El caso de uso representa una funcionalidad demandada por un actor.
- Precondiciones: Se realiza una descripción de las condiciones que deben cumplirse para poder realizar una operación.
- Poscondiciones: Se realiza una descripción del estado en el que queda el sistema tras realizar una operación.
- Escenario: Se realiza una descripción básica de las acciones que se ejecutaran paso a paso en el caso de uso.

3.2.3.3. Diagrama de los casos de uso

A continuación se mostrará un diagrama con los casos de uso identificados y la relación existentes entre los mismos y entre los actores identificados en el sistema 3.1:

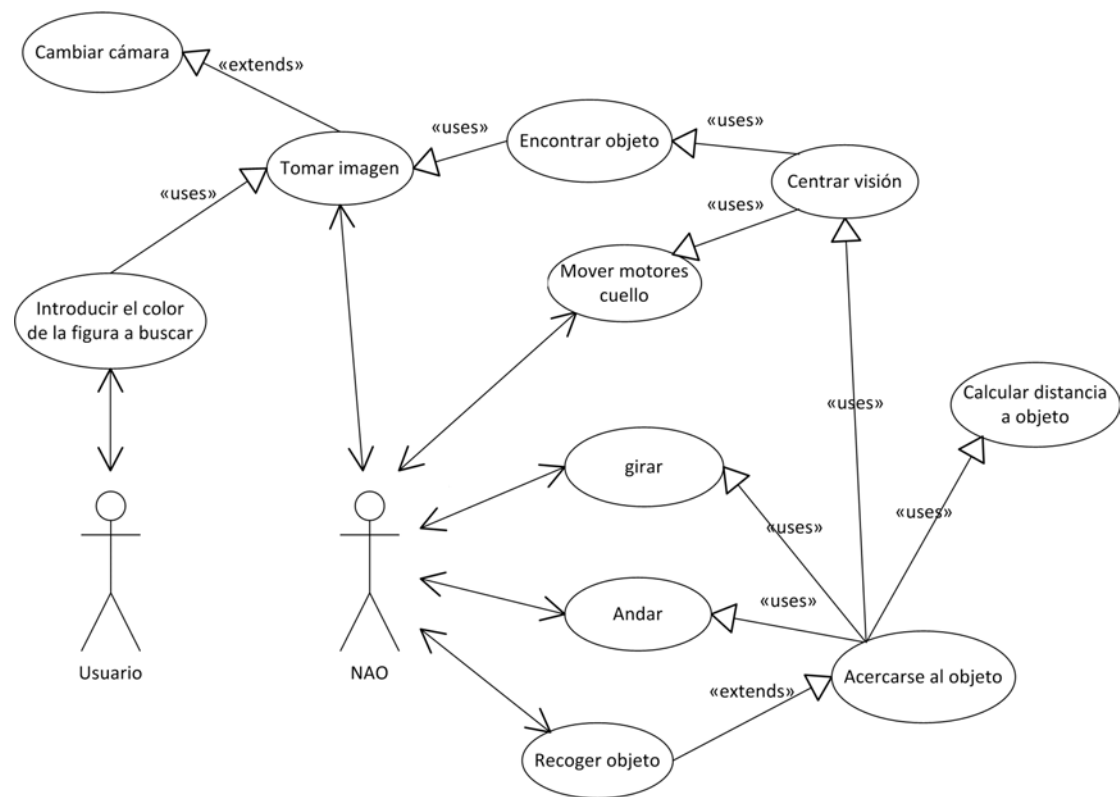


Figura 3.1: Diagrama de casos de uso.

3.2.3.4. Descripción textual de los casos de uso

A continuación se muestran los casos de uso identificados:

CU-001	
Nombre	Introducir el color de la figura a buscar
Actores	Usuario
Precondiciones	Haber iniciado el sistema para captar el objeto a buscar Tener el objeto colocado a 50cm del robot Estar el objeto completamente visible desde la posición del robot
Postcondiciones	Se guardan los atributos de color del objeto seleccionado
Escenario	1.- Se toma una imagen con la cámara del robot (CU-002. Tabla 3.2) 2.- El sistema procesa la imagen y se muestran los objetos encontrados al usuario 3.- El usuario selecciona la imagen que se quiere buscar

Tabla 3.1: Caso de uso 001

CU-002	
Nombre	Tomar imagen
Actores	Robot
Precondiciones	Tener el sistema conectado al robot La cámara del robot debe estar operativa
Postcondiciones	Se almacenará en la memoria del sistema la imagen del entorno
Escenario	1.- El sistema envía una petición al robot para pedir una captura de la cámara 2.- El robot envía una captura de la cámara seleccionada 3.- El sistema transforma la imagen a formato IplImage

Tabla 3.2: Caso de uso 002

CU-003	
Nombre	Cambiar cámara
Actores	Robot
Precondiciones	Tener el sistema conectado al robot Las cámaras del robot deben estar operativas
Postcondiciones	Se cambiará la cámara seleccionada para tomar una imagen
Escenario	1.- El sistema envía una petición al robot para cambiar la cámara con la que se tomará la imagen 2.- El robot selecciona la cámara nueva para tomar una imagen

Tabla 3.3: Caso de uso 003

CU-004	
Nombre	Mover motores cuello
Actores	Robot
Precondiciones	Tener el sistema conectado al robot
Postcondiciones	El motores del cuello del robot se moverán a los angulos que se determinen
Escenario	1.- El sistema envía una petición al robot para mover los motores del cuello a unos angulos determinados 2.- El robot posiciona los angulos del cuello a la posición indicada

Tabla 3.4: Caso de uso 004

CU-005	
Nombre	Centrar visión
Actores	Robot
Precondiciones	Tener el sistema conectado al robot Haber encontrado un objeto con la cámara del robot
Postcondiciones	El objeto permanecerá centrado en la imagen captada por la cámara del robot
Escenario	1.- El sistema calculara la distancia del objeto encontrado al centro de la imagen 2.- El sistema calcula los grados que han de moverse los motores del cuello del robot para que el objeto quede centrado en la imagen 3.- Se enviara una señal al robot para mover el cuello mediante el caso de uso CU-004. Tabla 3.4 3.- El robot realizara el movimiento del cuello mediante los motores indicados

Tabla 3.5: Caso de uso 005

CU-006	
Nombre	Calcular distancia a objeto
Actores	Robot
Precondiciones	Tener el sistema conectado al robot Tener la figura centrada en la visión de la cámara del robot
Postcondiciones	Se calcula la distancia al objeto centrado en la imagen
Escenario	1.- El sistema solicita al robot los ángulos del cuello 2.- El robot envía los valores de los motores solicitados 3.- El sistema calcula la distancia al objeto

Tabla 3.6: Caso de uso 006

CU-007	
Nombre	Andar
Actores	Robot
Precondiciones	Tener el sistema conectado al robot Indicar la distancia que se quiere andar
Postcondiciones	El robot andará la distancia indicada
Escenario	1.- El sistema envía una petición al robot para andar una distancia determinada 2.- El robot andará la distancia marcada

Tabla 3.7: Caso de uso 007

CU-008	
Nombre	Girar
Actores	Robot
Precondiciones	Tener el sistema conectado al robot Indicar el ángulo que se desea que el robot gire
Postcondiciones	El robot se posicionará con el ángulo indicado
Escenario	1.- El sistema envía una petición al robot para girar con un determinado ángulo 2.- El robot girará con respecto al ángulo indicado

Tabla 3.8: Caso de uso 008

CU-009	
Nombre	Acercarse al objeto
Actores	Robot
Precondiciones	Tener el sistema conectado al robot Tener la figura centrada en la visión de la cámara del robot
Postcondiciones	El robot estará en la posición indicada para recoger el objeto
Escenario	1.- El sistema calcula la posición del objeto en referencia al robot 2.- Se envía una señal al robot para girar y posicionarse frente al objeto mediante el caso de uso CU-008. Tabla 3.8 3.- Se envía una señal al robot para andar la distancia indicada mediante el caso de uso CU-007. Tabla 3.7

Tabla 3.9: Caso de uso 009

CU-010	
Nombre	Encontrar un objeto
Actores	Robot
Precondiciones	Tener el sistema conectado al robot
Postcondiciones	El sistema enconcontrará un objeto con un color dentro de un rango establecido
Escenario	<p>1.- El sistema tomará una imagen mediante el caso de uso CU-002. Tabla 3.2</p> <p>2.- El sistema encontrará los objetos dentro de la imagen que cumplan unas condiciones de color</p> <p>3.- El sistema seleccionará el objeto con mas parecido al color establecido</p>

Tabla 3.10: Caso de uso 010

CU-011	
Nombre	Recoger objeto
Actores	Robot
Precondiciones	<p>Tener el sistema conectado al robot</p> <p>Haberse acercado correctamente al objeto que se quiere recoger</p>
Postcondiciones	El robot cogerá un objeto
Escenario	<p>1.- El sistema enviará una señal para recoger el objeto</p> <p>2.- El robot realizara el comportamiento de recogida del objeto y volverá a la posición inicial</p>

Tabla 3.11: Caso de uso 011

3.2.4. Especificación de requisitos

3.2.4.1. Descripción de los atributos de los requisitos

Para la realización de la descripción textual de los distintos requisitos que han sido identificados, se han seleccionado una serie de atributos que describen cada uno de los requisitos. A continuación se realiza una descripción del significado de cada uno de los atributos utilizados para su descripción:

- **Nombre:** Identificación corta del requisito.
- **Código:** Identificación unívoca abreviada del requisito, se construye mediante el código del requisito seguido de un - y de tres dígitos. Los requisitos serán divididos en funciones y no funcionales y sus códigos son RF para los requisitos funciones y RNF para los requisitos no funcionales. Por ejemplo RF-001.
- **Descripción:** Se realiza una descripción básica del requisito que ha sido identificado.
- **Fuente:** Indica a través de que fuente ha sido identificado el requisitos. Normalmente este valor se corresponderá con uno o varios códigos de los casos de uso.
- **Necesidad:** Determina el grado de implementación del requisito. Los valores que puede tomar este atributo son los siguientes:
 - **Esencial:** El requisito tiene que ser implementado.
 - **Deseable:** Es preferible implementar el requisito, pero no es obligatorio.
 - **Opcional:** El requisito se podrá implementar, pero no es importante ni obligatorio.

- **Prioridad:** Define la importancia del requisito, de forma que permita definir el orden en el cual serán incluido en el proceso de diseño y el orden de implementación. Los valores que puede tomar este atributo son los siguientes:
 - **Alta:** El requisito debe ser implementado en las fases iniciales del desarrollo.
 - **Media:** El requisito debe ser implementado una vez que hayan sido implementados los requisitos de prioridad alta.
 - **Baja:** El requisito debe ser implementados en las fases finales del desarrollo. Estos requisitos no influirán en el correcto funcionamiento del sistema.
- **Estabilidad:** Define la estabilidad del requisitos durante la vida útil del software. Esto implica si el requisito podrá ser o no modificado durante el ciclo del vida. Los valores que puede tomar este atributo son los siguientes:
 - **Estable:** El requisito no puede variar durante el ciclo de vida del sistema.
 - **Inestable:** El requisito puede variar a lo largo de la ciclo de vida del sistema.
- **Verificabilidad:** Define el grado de verificabilidad de un requisito, es decir indica en qué grado es posible comprobar que el requisito se ha incorporado en el sistema desarrollado. Los valores que puede tomar este atributo son los siguientes:
 - **Alta:** Se puede verificar que el requisito ha sido implementado en el sistema. Este tipo de requisitos se corresponden con las funcionalidades básicas del sistema.
 - **Media:** Se puede verificar que el requisito ha sido implementado en el sistema. Pero requiere de una comprobación compleja o del código fuente del sistema.

- Baja: Es difícil verificar si el requisito ha sido implementado en el sistema o en algunos casos no es posible.
- Descripción: Descripción extendida de la funcionalidad del requisito.

3.2.4.2. Descripción textual de los requisitos

Se ha realizado una división de los requisitos encontrados entre los requisitos funcionales y no funcionales. Los requisitos funcionales representan las funcionalidades que el sistema será capaz de realizar mientras que los requisitos no funcionales representan características que puedan limitar el sistema de alguna manera. A continuación se listan todos los requisitos funcionales:

Tomar imagen			
Código	RF-001	Fuente	CU-002
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El sistema debe ser capaz enviar una señal al robot para captar una imagen con la cámara		

Tabla 3.12: Requisito funcional 001

Cambiar cámara			
Código	RF-002	Fuente	CU-003
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de enviar una señal al robot para seleccionar una cámara diferente		

Tabla 3.13: Requisito funcional 002

Buscar objetos (sin color)			
Código	RF-003	Fuente	CU-001
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de encontrar los objetos presentes en una imagen sin necesidad de conocer su color		

Tabla 3.14: Requisito funcional 003

Selección de un objeto para buscar			
Código	RF-004	Fuente	CU-001
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de mostrar las figuras encontradas y guardar la información relativa al objeto que seleccione el usuario		

Tabla 3.15: Requisito funcional 004

Buscar objetos (con color)			
Código	RF-005	Fuente	CU-007
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de encontrar los objetos presentes en una imagen que sean de un determinado color		

Tabla 3.16: Requisito funcional 005

Centrar visión en un objeto			
Código	RF-006	Fuente	CU-004
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de enviar una señal al robot para girar la cabeza y dejar el objeto encontrado en el centro de la imagen		

Tabla 3.17: Requisito funcional 006

Andar			
Código	RF-007	Fuente	CU-006
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El sistema debe ser capaz enviar una señal al robot para andar una determinada distancia		

Tabla 3.18: Requisito funcional 007

Localización de objetos			
Código	RF-008	Fuente	CU-007
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	El sistema debe ser capaz de cubrir los 360 grados de visión hasta localizar el objeto seleccionado		

Tabla 3.19: Requisito funcional 008

Calcular distancia (Mesa)			
Código	RF-009	Fuente	CU-005
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de calcular la distancia hasta la mesa. La mesa sobre la que estará posicionada el objeto tendrá siempre las mismas características de tamaño y color		

Tabla 3.20: Requisito funcional 009

Calcular distancia (Objeto)			
Código	RF-010	Fuente	CU-005
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Estable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de calcular la distancia hasta el objeto que esta posicionado encima de la mesa		

Tabla 3.21: Requisito funcional 010

Información de los objetos encontrados			
Código	RF-011	Fuente	CU-007
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema guardara información de los elementos encontrados en la imagen tales como: color, anchura, altura, etc.		

Tabla 3.22: Requisito funcional 011

Posicionar al robot frente al objeto			
Código	RF-012	Fuente	CU-006
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	El sistema debe ser capaz de colocarse adecuadamente para recoger el objeto. La posición del robot respecto al objeto debe ser aproximadamente 16 cm respecto a la perpendicular y 12 cm respecto a la horizontal.		

Tabla 3.23: Requisito funcional 012

Recogida del objeto			
Código	RF-013	Fuente	CU-008
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El sistema debe de ser capaz de ejecutar un comportamiento con el robot para recoger un objeto situado en una posición determinada.		

Tabla 3.24: Requisito funcional 013

Buscar objeto en el entorno			
Código	RF-014	Fuente	CU-010
Necesidad	Media	Prioridad	Media
Estabilidad	Estable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de realizar una búsqueda en el entorno cercano si no se ha encontrado el objeto deseado en la imagen tomada.		

Tabla 3.25: Requisito funcional 014

Recolocar robot frente al objeto.			
Código	RF-015	Fuente	CU-008
Necesidad	Media	Prioridad	Media
Estabilidad	Estable	Verificabilidad	Media
Descripción	El sistema debe ser capaz de enviar una señal al robot para girar posicionandose frente al objeto encontrado.		

Tabla 3.26: Requisito funcional 015

A continuación se listan todos los requisitos no funcionales:

Robot utilizado			
Código	RNF-001	Fuente	—
Necesidad	Esencial	Prioridad	Alta
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El robot utilizado en el sistema será el robot NAO.		

Tabla 3.27: Requisito no funcional 001

Tiempo de ejecución			
Código	RNF-002	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El tiempo de ejecución no deberá ser superior a 5 minutos.		

Tabla 3.28: Requisito no funcional 002

Alta modularización			
Código	RNF-003	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El diseño del sistema deberá ser modular para permitir ampliar la funcionalidad de una manera sencilla.		

Tabla 3.29: Requisito no funcional 003

Facilidad de uso			
Código	RNF-004	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema deberá ser fácilmente usable.		

Tabla 3.30: Requisito no funcional 004

Equipo de usuario			
Código	RNF-005	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El sistema se ejecutará sobre un ordenador.		

Tabla 3.31: Requisito no funcional 005

Equipo de comunicación			
Código	RNF-006	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	Para la comunicación se utilizará un ruoter wifi.		

Tabla 3.32: Requisito no funcional 006

Conexión			
Código	RNF-007	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	Se conectará el ordenador con el robot antes de la ejecución del sistema a la dirección IP configurada para ese robot.		

Tabla 3.33: Requisito no funcional 007

Distancia del objeto			
Código	RNF-008	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El objeto a buscar deberá estar a una distancia máxima de 2 metros.		

Tabla 3.34: Requisito no funcional 008

Posición objeto			
Código	RNF-009	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El objeto deberá estar colocado a 31cm de altura sobre una caja de cartón.		

Tabla 3.35: Requisito no funcional 009

Visibilidad			
Código	RNF-010	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El objeto deberá ser totalmente visible desde la posición del robot.		

Tabla 3.36: Requisito no funcional 010

Color de la caja			
Código	RNF-011	Fuente	—
Necesidad	Deseable	Prioridad	Media
Estabilidad	Inestable	Verificabilidad	Media
Descripción	La caja de cartón deberá tener la parte superior de color blanco para no dificultar la búsqueda del objeto.		

Tabla 3.37: Requisito no funcional 011

3.3. Diseño del sistema

El objetivo del proceso de Diseño del Sistema de Información es la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

3.3.1. Arquitectura del sistema

En este apartado se mostrará la arquitectura del sistema y su descomposición en distintos componentes. Los componentes principales del sistema desarrollado se han basado en una arquitectura de capas top-down en la que encontramos:

- Capa del robot: Esta capa es la encargada de realizar peticiones o accesos a elementos del robot.
- Capa procesamiento de imágenes: En esta capa se realizarán las operaciones relacionadas con el tratamiento de imágenes. En esta capa se utilizará la librería OpenCv.
- Capa procesamiento de colores: Esta capa lleva a cabo operaciones con imágenes únicamente a nivel de píxel. En esta capa se realizarán todas las operaciones relacionadas con la detección y el reconocimiento de colores.

Cada capa puede enviar solicitudes de servicio a elementos de la capa inmediatamente inferior. Esta arquitectura se puede ver en la figura 3.2. Así mismo si se necesita realizar una solicitud a una capa más baja se produce un comportamiento en cascada enviando solicitudes por cada capa hasta llegar a la capa solicitada.

La principal razón para elegir esta arquitectura es la modularidad que ofrece la disposición formando diferentes capas. Cada capa es independiente de las otras y solo conoce la comunicación realizada con las capas inferiores, por lo tanto no

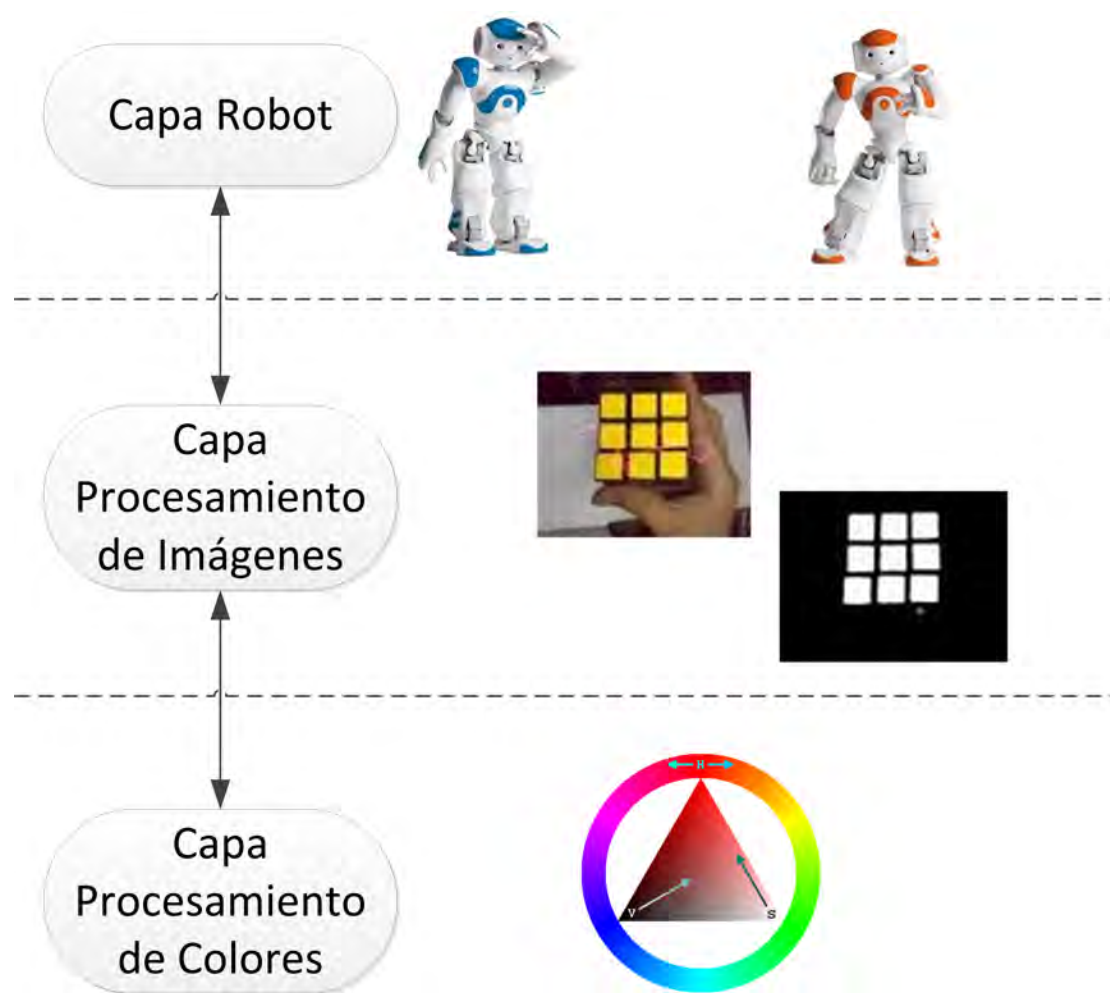


Figura 3.2: Modelo de arquitectura Top-Down.

es necesario conocer el funcionamiento interno. Con esta arquitectura por capas el sistema consigue una gran flexibilidad, ya que al estar distribuido en módulos independientes estos pueden ser intercambiados por otros fácilmente sin tener que modificar el total del sistema. Por ejemplo, si se desea cambiar el robot por otro más avanzado solo es necesario cambiar la capa que se ocupa de interactuar con el robot sin necesidad de modificar las otras capas.

3.3.2. Descripción general del sistema

En este apartado se realiza una descripción general del funcionamiento del sistema. Para ello se identificaran las fases por las que se va pasando hasta conseguir el resultado final. La descripción del funcionamiento del sistema se presenta en el apartado *3.2.1 Descripción de las características funcionales*. En la imagen 3.3 se muestran las etapas en las que esta dividido el funcionamiento del sistema. A continuación se describe cada una de estas etapas:

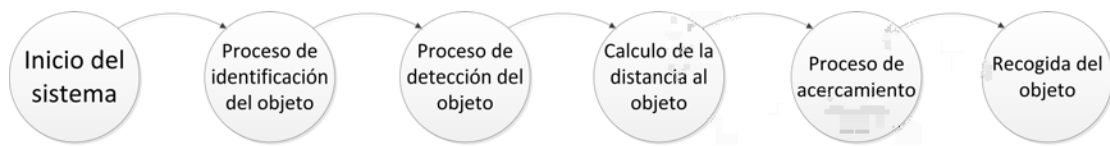


Figura 3.3: Diagrama del funcionamiento del sistema.

3.3.2.1. Proceso de identificación del objeto

Los procesos principales que se realizan en esta fase son:

- **Tomar imagen:** Se toma una captura de imagen con la cámara del robot mediante el módulo de visión ALVideoDevice englobada dentro del framework NAOqi. Este módulo es el encargado de gestionar las entradas de video a través de los dispositivos de entrada presentes en el robot Nao. Para el posterior procesamiento las imágenes capturadas se convertirán a la estructura IplImage de la librería OpenCv.
- **Procesar imagen en busca de objetos:** Se recorre la imagen en busca de agrupaciones de colores. Cada agrupación de color se trata como un objeto independiente. Una vez procesada la imagen se guardará un vector con los objetos encontrados.
- **Mostrar objetos al usuario:** Se mostrará al usuario los objetos captados y se le pedirá seleccionar uno para su posterior búsqueda. La selección de

la figura se realizará mediante los botones de la cabeza del robot. Para el objeto seleccionado se guardará el valor medio del color en formato HSV y se guardará para realizar la búsqueda.

Como se puede observar en la imagen 3.4 para realizar este proceso se realiza un procedimiento de detección de colores para después segmentar la imagen en las zonas en las que se concentran las mismas características de color.

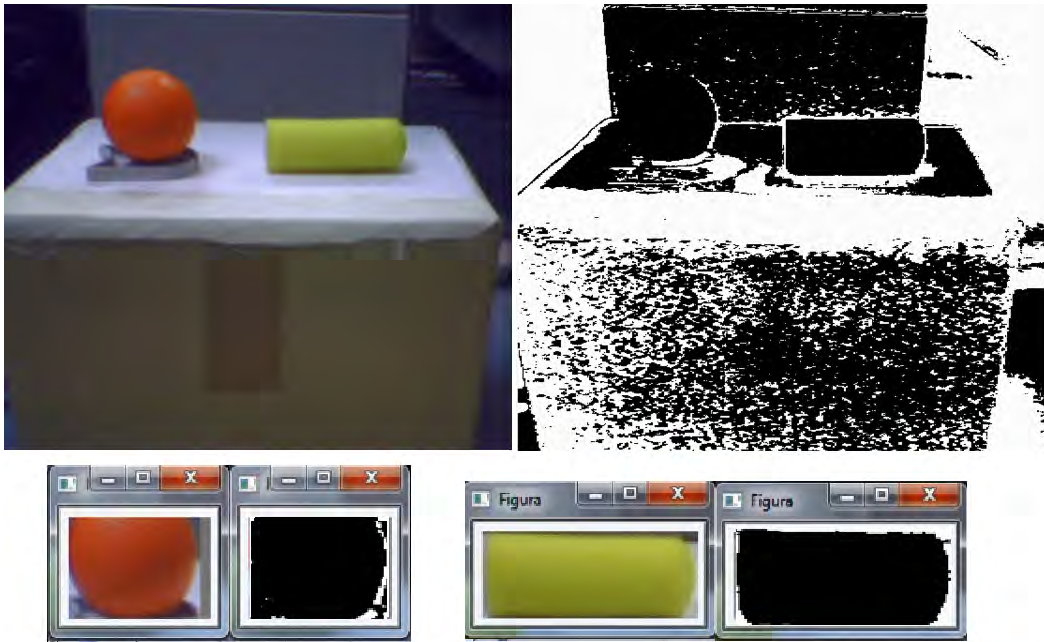


Figura 3.4: Procesamiento de imagenes para la detección de objetos.

Este proceso podrá ser repetido para seleccionar la mesa sobre la que estará apoyada el objeto y guardar sus valores del color. En el caso de que no se introduzca este valor se utilizaran los valores por defecto.

3.3.2.2. Proceso de detección del objeto

Una vez introducido el color a buscar se deberá colocar el escenario de ejecución. Al terminar se pulsará cualquier tecla del ordenador para comenzar a buscar el objeto y el robot tomará la posición inicial (definida en el apéndice C.2) capturando una imagen del entorno. Se analizará la imagen buscando agrupaciones de

colores similares al identificado en la fase anterior. Una vez procesada la imagen y detectados todos los objetos se aplicara una función de similitud seleccionando el objeto con mayor parecido.

3.3.2.3. Cálculo de la distancia a la figura

Los procesos principales que se realizan en esta fase son:

- **Centrar la cámara en la figura seleccionada:** Se mandara una señal al robot para mover el cuello en el eje x e y mediante el módulo ALMotion englobado dentro del framework de NAOqi. Este módulo facilita el movimiento del robot Nao asignando ángulos a un motor o conjunto de motores [C.1]. En esta fase se pretende dejar el objeto centrado en la visión del robot. Para ello se calcula la distancia, en el eje x e y, del punto medio de la base de la figura con respecto al centro de la imagen y se utiliza una función para transformar esta distancia en píxeles de la imagen a ángulos del cuello.

A continuación se muestra un ejemplo de la realización de este proceso en un caso real. En este ejemplo se ve como se produce el centrado de la mesa en la imagen recogida por el robot con el fin de calcular la distancia posteriormente. En la imagen 3.5 se observa el proceso de centrado del objeto desde el momento inicial en el que se visualiza hasta que este queda en el centro de la imagen como se ha descrito anteriormente. Para este ejemplo se han realizado dos interacciones. En cada interacción se ha realizado el reconocimiento de la figura en la imagen, se ha calculado el punto de la figura para realizar el centrado, se obtiene la distancia de ese punto al centro de la imagen y por último se transforma esa distancia a los ángulos que se deben adoptar con el fin de que la figura quede totalmente centrada. Como se observa en el ejemplo en la segunda iteración realizada se ha conseguido centrar la figura satisfactoriamente.



Figura 3.5: Centrado de la cámara en una figura.

- **Calculo de la distancia:** Cuando la figura está completamente centrada en la imagen recogida por el robot se utiliza el teorema del seno para calcular a que distancia se halla la figura.

3.3.2.4. Movimiento del robot

Se realiza el movimiento del robot para posicionarlo frente al objeto a una distancia fija. Para realizar el proceso de recogida correctamente la posición final del robot con respecto al objeto debe ser de 18 cm en el eje X y de 8 cm en el eje Y. Debido a esto es necesario modificar el valor resultante del cálculo de la distancia del sistema para dirigirse a la posición desde la que comenzará el proceso de recogida. Debido a que la recogida se ha implementado únicamente mediante la mano izquierda del robot la posición del robot se situará siempre a 18 cm a la derecha del objeto sobre el eje X. En la figura 3.6 se observa una recreación gráfica de la posición en la que debe situarse el robot.

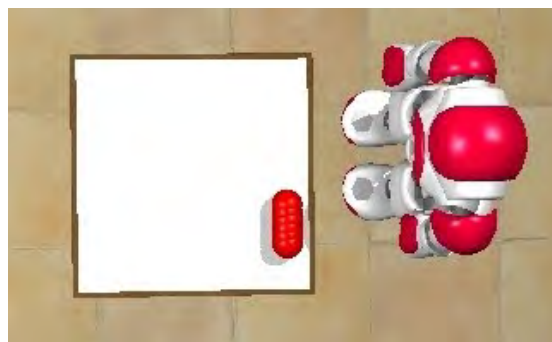


Figura 3.6: Posición necesaria para el proceso de recogida.

El proceso de movimiento se realiza mediante el método `walkTo` incluido en el módulo `ALMotion`. Para asegurar que el objeto sea recogido correctamente se volverán a realizar todos los pasos a partir de la fase *3.3.2.2 Búsqueda de la figura introducida* para evitar desvíos producidos en el proceso de andar del robot o un mal cálculo de la distancia.

3.3.2.5. Recogida del objeto

Para la recogida del objeto se ha creado un comportamiento definiendo una secuencia de posiciones del brazo del robot. Este comportamiento permite al robot, una vez se halla en la posición indicada, extender el brazo hasta alcanzar el objeto y cerrar la mano sujetándolo con fuerza para llevar por último el brazo a su posición inicial.

3.3.3. Descripción de componentes

En este apartado se va a proceder a dar una descripción detallada del funcionamiento de cada componente. Cada componente identificado está englobado en una única clase. A continuación se va a realizar una descripción de cada una de ellas a través del funcionamiento de sus métodos más destacados. En la figura 3.7 vienen indicados los componentes junto a los métodos de cada uno:

3.3.3.1. Módulo robot

En esta sección se realizará una descripción de la funcionalidad implementada en este módulo. Este módulo es la encargada de realizar las operaciones relacionadas con el robot. Este módulo se corresponde con la clase implementada en el fichero `Buscar.cpp`.

3.3.3.1.1. Proceso de aproximación a la mesa

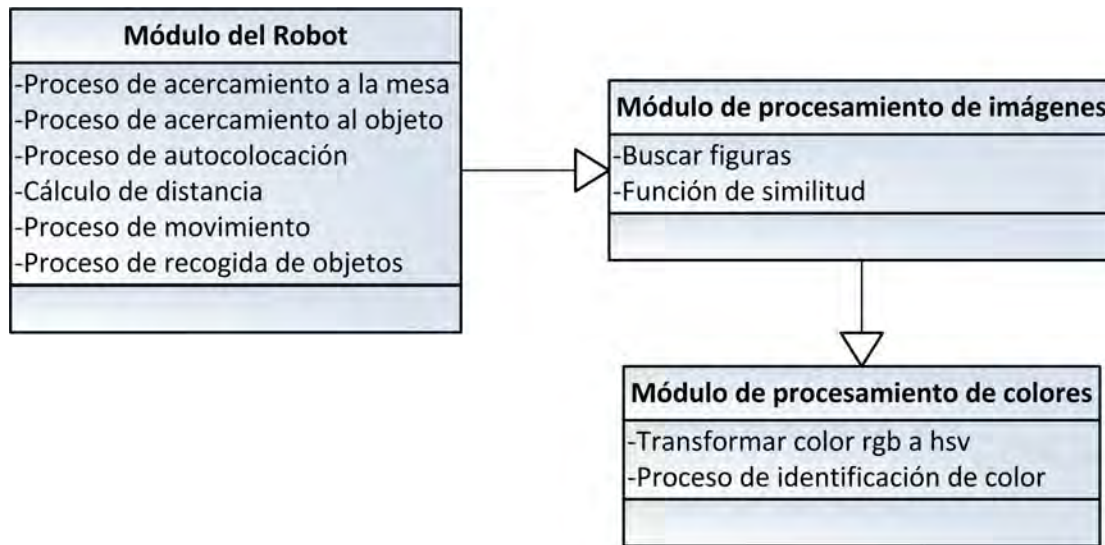


Figura 3.7: Diagrama de componentes.

La funcionalidad de este método es detectar la mesa utilizada para colocar el objeto que se quiere buscar y acercarse a ella quedándose a una distancia de 20 cm aproximadamente. Se ha comprobado que en esta distancia se alcanza el máximo índice de aciertos en el proceso de calcular la distancia y recoger el objeto. Este procedimiento se realiza en dos ocasiones, cuando se ha encontrado la mesa pero no el objeto, o cuando se ha encontrado el objeto pero está demasiado lejos como para calcular la distancia correctamente. En estas dos ocasiones se pretende el mismo objetivo, acercarse a la mesa para tomar una imagen más cercana de la superficie y así poder localizar el objeto con mayor precisión o confirmar la ausencia del mismo.



Figura 3.8: Etapas de aproximación a la mesa.

En el diagrama 3.8 se muestran las etapas de este procedimiento. A continuación se realiza una descripción de cada etapa con más detalle:

1. **Tomar imagen y buscar mesa:** Se toma una imagen con la cámara su-

perior del robot y se procesa en busca del tono de color seleccionado para la mesa. Si en el escenario no encuentra el objeto mesa se enviará una señal al robot para girar 45 grados en sentido contrario a las agujas del reloj. Después de realizar el movimiento se vuelve a repetir el mismo proceso para buscar la mesa.

2. **Calcular distancia a la mesa:** Una vez encontrado el objeto se calcula la distancia al mismo mediante el método *3.3.3.1.4*.
3. **Realizar movimiento de acercamiento a la mesa:** Se realizará el movimiento hacia la mesa con el fin de situarse a menos de 30 cm de la misma. Aunque la distancia óptima para el proceso de recogida del objeto es de 20 cm se le da un margen de 10 cm al acercamiento a la mesa debido a las imprecisiones del movimiento del robot.
4. **Comprobar que la distancia actual sea la adecuada:** Se realizara una comprobación con la cámara inferior del robot para asegurar que la distancia a la mesa es la adecuada antes de empezar a buscar el objeto. El cálculo de esta comprobación se puede ver en el apartado *3.3.3.1.4*.

3.3.3.1.2. Proceso de aproximación al objeto

Este proceso tiene como objetivo localizar el objeto a recoger y aproximar el robot a una distancia desde la que se pueda recoger satisfactoriamente. Este método se ejecuta cuando la mesa ha sido detectada a una distancia inferior a 30 cm.



Figura 3.9: Etapas de aproximación al objeto.

En el diagrama 3.9 se muestran la etapas de este procedimiento. A continuación se realiza una descripción de cada etapa con más detalle:

1. **Tomar imagen y buscar objeto:** Se toma una imagen con la cámara inferior del robot y se procesa en busca del tono de color seleccionado para el objeto a recoger. Si en la imagen captada no se encuentra el objeto se enviará una señal al robot para girar el cuello y tomar una nueva imagen a cada lado para aumentar el espacio de búsqueda.
2. **Calcular distancia al objeto:** Una vez encontrado el objeto se calcula la distancia al mismo mediante el metodo *3.3.3.1.4 Calcular distancia*.
3. **Realizar movimiento de acercamiento al objeto:** Se realizará el movimiento hacia el objeto con el fin de situar al robot a una distancia de 15 cm en el eje x y 8 cm en el eje y.
4. **Comprobar que la distancia actual sea la adecuada:** Se realizará una comprobación con la cámara inferior del robot mediante el método descrito en el apartado *3.3.3.1.4* para asegurar que la distancia al objeto es la adecuada para realizar la recogida del mismo. Si la distancia al objeto no es la correcta se volverá a ejecutar el método para corregir la desviación producida.

3.3.3.1.3. Proceso de autocolocación de visión

Centrar la visión en la figura tiene como objetivo colocar la figura en el área central de la imagen para poder calcular la distancia. El método desarrollado para calcular la distancia (explicado en la sección *3.3.3.1.4*) necesita que el objeto que se toma como referencia se encuentre lo más centrado posible en la imagen. Debido a este proceso de centrado de la figura, es necesario realizar sucesivos movimientos sobre los motores del cuello del robot. La necesidad de conectarse al robot para realizar tal movimiento y comprobar de nuevo la posición del objeto hasta que se

realiza correctamente el centrado provoca un gasto de tiempo considerable, por ello se ha intentado que este proceso se realice con el mínimo número de repeticiones posibles. Para ello se han calculado los ángulos del cuello de la siguiente forma:

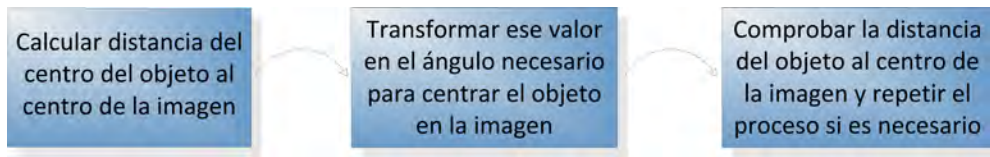


Figura 3.10: Etapas de aproximación al objeto.

En el diagrama 3.10 se muestra como se ha realizado el proceso de centrado de la visión en el objeto. A continuación se realiza una descripción de cada etapa con más detalle:

1. Se calcula el punto medio del eje horizontal del objeto y el punto mas bajo del eje vertical (Figura 3.11)

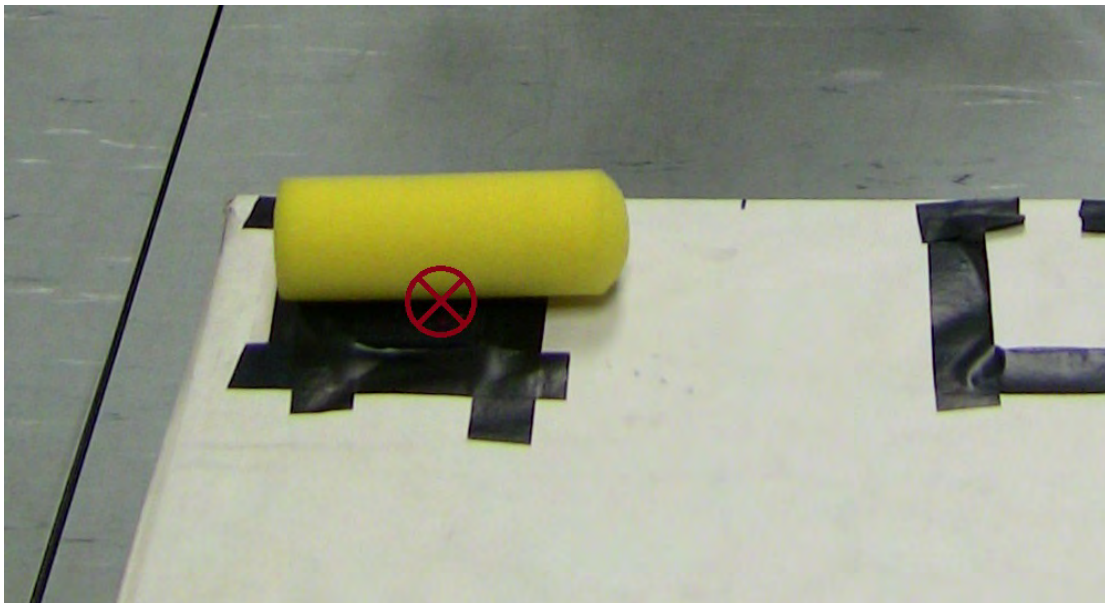


Figura 3.11: Punto de referencia para centrar la visión.

2. Se calculan los ángulos necesarios de los motores HeadYaw y Head Pitch para que el objeto quede centrado mediante las ecuaciones 3.1 y 3.2. En el Anexo C.1 se puede observar una lista con los motores del robot NAO.

$$AnguloX = \alpha + \frac{x - \frac{Width}{2}}{-700} \quad (3.1)$$

AnguloX = Ángulo resultante para el motor HeadYaw

α = Ángulo actual del motor HeadYaw

x = Distancia del punto medio del objeto al punto medio de la imagen en el eje x

Width = Anchura de la imagen en píxeles

$$AnguloY = \beta + \frac{y - \frac{High}{2}}{700} \quad (3.2)$$

AnguloY = Ángulo resultante para el motor Head Pitch

β = Ángulo actual del motor Head Pitch

y = Distancia del punto más bajo del objeto al punto medio de la imagen en el eje y

High = Altura de la imagen en píxeles

3. Se comprueba la nueva posición del objeto en la figura. Si la distancia euclídea al centro de la imagen es mayor que 20 se realiza otra iteración.

Este proceso de centrado garantiza que el objeto quede correctamente centrado necesitando realizar el movimiento del cuello entre dos y tres veces.

3.3.3.1.4. Cálculo de la distancia

Cuando se ha encontrado una figura en la imagen y se ha centrado convenientemente se calcula la distancia a la misma mediante el teorema del seno. El teorema del seno define que, en todo triángulo, los lados son proporcionales a los senos de los ángulos opuestos, como puede verse en la figura 3.12.

Mediante este teorema podemos calcular la distancia a un objeto conociendo los siguientes datos:

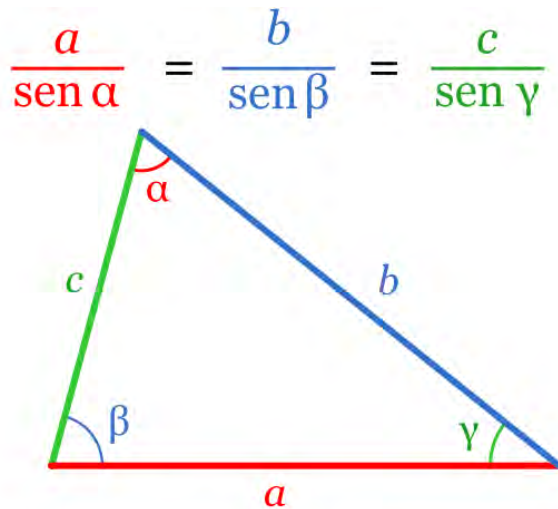


Figura 3.12: Teorema del seno.

- **Altura cámara:** La altura de la cámara al suelo varía dependiendo de la cámara utilizada para captar el objeto. Para la cámara situada en la frente se tiene un valor de 47.5 cm mientras que la cámara situada en la barbilla se encuentra a 43 cm del suelo. Debido a que esta distancia al suelo es únicamente válida en la posición base (mostrada en el apéndice C.2) es necesario recuperar la posición cada vez que se va a centrar la visión en la figura.
- **Ángulos del cuello de robot:** Los ángulos del cuello corresponden a los valores de los motores HeadYaw y Head Pitch. Junto a estos valores hay que realizar una corrección del ángulo de la cámara usada. Esta corrección es necesaria para calcular el ángulo del cuello a partir de la horizontal. Como se puede ver en la figura 3.13 la corrección de la cámara superior es de 1.2° y de la cámara inferior de 39.7°.
- **Distancia al suelo de la figura:** Este valor depende de si buscamos la mesa o el objeto que queremos recoger. La mesa siempre va a tener una distancia de 0 ya que estará apoyada sobre el suelo, mientras que para el objeto a recoger usaremos una distancia por defecto de 31cm. Este valor se ha seleccionado debido a la altura de la mesa utilizada para llevar a cabo

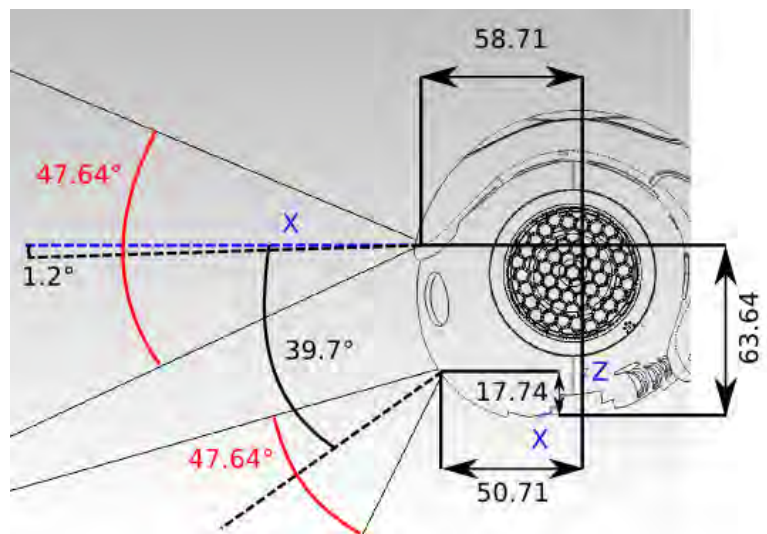


Figura 3.13: Hardware cámara.

las pruebas.

Mediante la utilización de estos datos se realiza un cálculo de la distancia tanto en el eje X como en el eje Y. Para ello se utiliza el teorema del seno construyendo un triángulo rectángulo entre el objeto y el robot definiendo el ángulo de uno de los lados como el valor del motor HeadYaw (β). Con ello se obtiene un triángulo en el que se conocen todos sus ángulos y sus dos catetos se corresponden a la distancia en el eje X y a la distancia en el eje Y cómo se puede observar en la imagen 3.14.

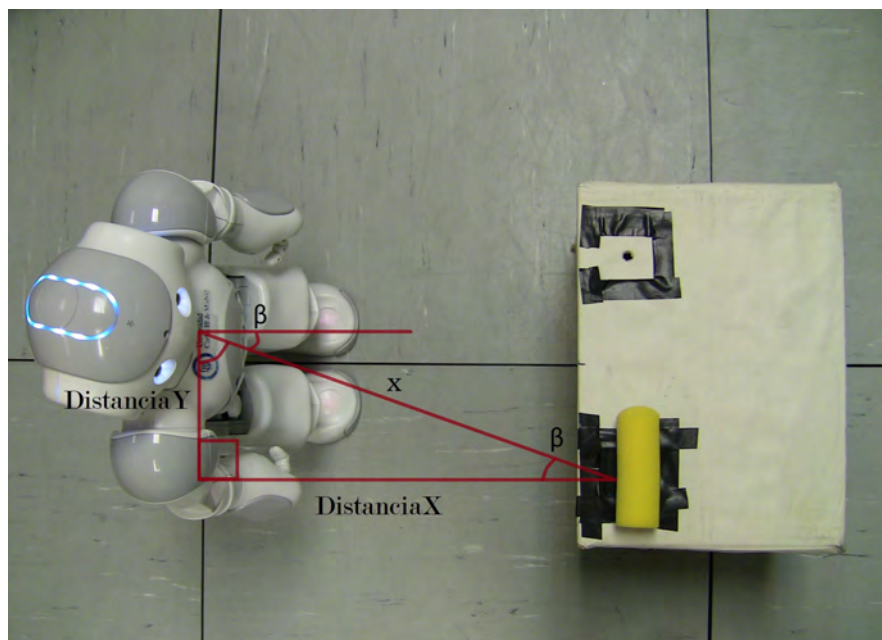


Figura 3.14: Aplicación del teorema del seno.

Para poder aplicar el teorema del seno y calcular la distancia del robot al objeto es necesario conocer al menos el valor de uno de los lados del triángulo. Por ello se ha realizado el cálculo del valor X utilizando nuevamente el teorema de seno, como se observa en la figura 3.15 esta vez construyendo un triángulo rectángulo entre la cámara del robot y el objeto definiendo el ángulo de uno de los lados como el valor del motor HeadPitch (α).

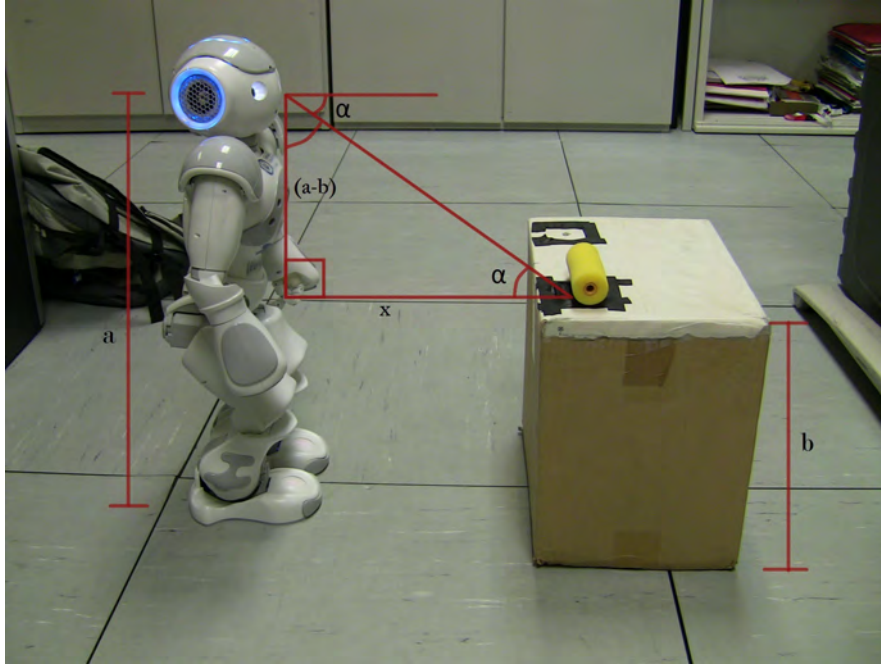


Figura 3.15: Aplicación del teorema del seno.

Por lo tanto para realizar el cálculo de la distancia tanto en el eje X como en el eje Y se utilizan las formulas 3.3 , 3.4 y 3.5.

$$x = \frac{(a - b) * \text{sen}(\frac{\pi}{2} - \alpha)}{\text{sen}(\alpha)} \quad (3.3)$$

$$\text{DistanciaX} = x * \text{sen}(\frac{\pi}{2} - \beta) \quad (3.4)$$

$$\text{DistanciaY} = x * \text{sen}(\beta) \quad (3.5)$$

3.3.3.1.5. Proceso de movimiento

Se realiza el comportamiento WalkTo incorporado en el módulo de movimiento del framework NAOqi. Para realizar el comportamiento de recogida adecuadamente se envia una señal al robot para andar y posicionarse a una distancia fija relativa al objeto. Esta distancia relativa al objeto es la necesaria para que al

extender el brazo del robot el objeto quede a la altura de los dedos y pueda ser agarrado.

3.3.3.1.6. Proceso de recogida de objetos

Se ejecuta el comportamiento de recogida del objeto. Este comportamiento está compuesto por 9 posiciones del brazo como se observa en la figura 3.16. Se ha implementado la recogida de los objetos únicamente con el brazo izquierdo del robot por comodidad, debido a que es indiferente realizar la recogida de uno u otro lado. Esta composición de movimientos permiten, situar el brazo sobre la mesa partiendo de la posición inicial, realizar un barrido hacia delante para asegurarse de recoger el objeto y volver a colocar el brazo en la posición inicial bordeando la mesa nuevamente. La recogida se ha implementado en forma de barrido para asegurar que se recoge la figura aun cuando la posición del robot respecto al objeto está ligeramente desviada. Una vez que se ha terminado el comportamiento, el brazo queda en la posición inicial dando la posibilidad al robot de que pueda moverse libremente sin complicaciones.

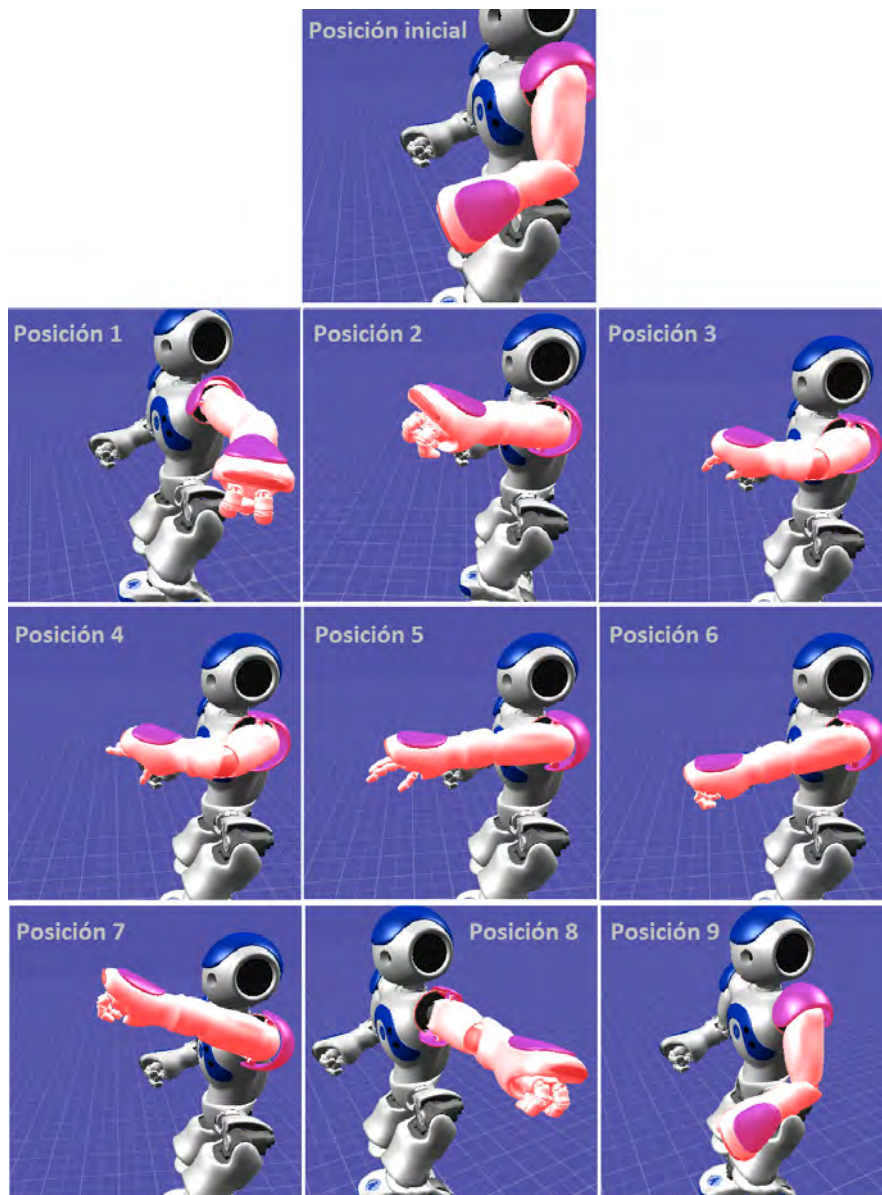


Figura 3.16: Movimientos de recogida.

3.3.3.2. Módulo de procesamiento de imágenes

Este módulo es el encargado de realizar el procesamiento y análisis de las imágenes. Las imágenes se han guardado en formato `IplImage` de la librería `OpenCV`. Esta estructura almacena todos los datos de la imagen de la que se utilizarán, la anchura, la altura, el número de canales y la matriz con los datos de la imagen. Este módulo se corresponde con la clase implementada en el fichero `Figura.cpp`.

3.3.3.2.1. Buscar figuras

Esta funcionalidad es la encargada de procesar las imágenes con el objetivo de localizar los objetos que encajen con unas características de color definidas. Este proceso se ha realizado buscando la mayor precisión para asegurar un buen comportamiento del sistema en tiempo de ejecución. Se ha hecho un gran hincapié en la eliminación del posible ruido presente en la imagen y en la correcta identificación de los colores detectados.

El procesado de la imagen se ha realizado de la siguiente forma:

1. Se realiza un procesado de la imagen recorriendo cada pixel de la imagen hasta encontrar el color que se está buscando.
2. Una vez se ha encontrado el color se crea una estructura figura con el objetivo de guardar los datos necesarios de esa figura. Los datos que se guardan de cada figura son:
 - a) Punto superior izquierdo de la zona de la imagen donde se encuentra la figura.
 - b) Punto inferior derecho de la zona de la imagen donde se encuentra la figura
 - c) Primer punto encontrado de la figura almacenada.
 - d) Último punto encontrado de la figura almacenada.
 - e) Punto central de la figura en la imagen original.
 - f) Altura en píxeles de la figura.
 - g) Anchura en píxeles de la figura.
 - h) Contador de píxeles del color buscado en la figura.
 - i) Valor medio de la tonalidad de la figura.
 - j) Valor medio de la saturación de la figura.
 - k) Valor medio del brillo de la figura.

3. A continuación se sigue recorriendo la imagen procesando los pixeles cercanos e introduciéndolos en la estructura si se corresponden con el color requerido.
4. Una vez que se ha recorrido toda la imagen se debe obtener un vector con todas las figuras (o regiones de colores) encontradas en la imagen que se correspondan con el color a buscar.
5. Para finalizar se realiza un post-procesado de las figuras encontradas borrando todas las que no se adapten con el objeto a buscar.

3.3.3.2.2. Función de similitud

Este proceso calcula la similitud que existe entre cada uno de los objetos detectados en la imagen con respecto al que se desea localizar. Este valor se calcula mediante la distancia euclídea entre los atributos de tono y saturación media de la figura buscada y la figura encontrada. En la imagen 3.17 se pueden observar los valores de los atributos de colores de alguna de las figuras detectadas.



Figura 3.17: Atributos de color de los objetos.

Como resultado se obtiene la distancia en un eje de coordenadas de dos dimensiones entre los puntos definidos por los atributos de color de cada figura. La función de similitud da como resultado un valor numérico que describe la similitud entre un objeto detectado y el objeto a buscar. De forma que cuanto menor es el valor obtenido tras aplicar la función de similitud, mayor es el grado de similitud entre los objetos. Con esta función nos aseguramos que en cada búsqueda realizada se encuentra el objeto deseado. La función utilizada es 3.6:

$$P = |T1 - T2| * 100 + |S1 - S2| \quad (3.6)$$

T1 = Valor del tono del color del objeto a buscar

T2 = Valor del tono del color del objeto encontrado

S1 = Valor de la saturación del color del objeto a buscar

S2 = Valor de la saturación del color del objeto encontrado

Para asegurarnos de que el objeto encontrado es exactamente el que se busca se le da un valor máximo al resultado obtenido de 500. Durante la realización de las experimentaciones se comprobó que con valores superiores a 500 se aprecia una clara diferencia entre las figuras procesadas. Por lo tanto todos los objetos encontrados que tengan un valor superior se eliminarán definitivamente.

3.3.3.3. Módulo de procesamiento de colores

Este módulo procesa la estructura de una imagen. Debido a las necesidades del trabajo se utiliza una composición de color en HSV para facilitar la detección del color. Este módulo se corresponde con la clase implementada en el fichero Color.cpp.

3.3.3.3.1. Transformar color rgb a hsv

Debido a que la imagen inicial se encuentra en formato RGB es necesario realizar una transformación a HSV para facilitar la tarea de reconocimiento de colores. Cada componente de la tupla HSV se calcula mediante una formula específica. Estas transformaciones se realizan a través de las ecuaciones 3.7, 3.8 y 3.9:

■ **Tono o Matiz:**

$$T = \begin{cases} 60^\circ \times (\frac{G' - B'}{\Delta} \text{ mód } 6) & , C_{\max} = R \\ 60^\circ \times (\frac{B' - R'}{\Delta} + 2) & , C_{\max} = G \\ 60^\circ \times (\frac{R' - G'}{\Delta} + 4) & , C_{\max} = B \end{cases}$$

(3.7)

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

■ **Saturación:**

$$S = \begin{cases} 0 & , \Delta = 0 \\ \frac{\Delta}{C_{\max}} & , \Delta \neq 0 \end{cases} \quad (3.8)$$

$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

■ **Valor o Luminosidad:**

$$V = C_{\max} \quad (3.9)$$

$$C_{\max} = \max(R', G', B')$$

Como resultado de estas formulas se obtiene una tupla de tres valores que se han normalizado de 0 a 240 para facilitar el tratamiento de los colores.

3.3.3.3.2. Proceso de identificación de color

Mediante este proceso se realiza una comparación de cada tupla de color HSV con el color que se buscar. Para ello se han creado un rango de valores entre los que tiene que estar para determinar que el color es el mismo que se desea buscar. Este

rango de valores se ha establecido comprobando el valor HSV de pixeles de una figura desde diferentes imágenes. Los rangos identificados son los siguientes:

- **Tonalidad:** Se ha establecido un rango de ± 5 para en el que se ha identificado que el color sigue siendo considerado el mismo que el buscado.
- **Saturación:** El valor de la saturación se muestra aproximadamente constante para una imagen pero puede ser considerablemente diferente al tomar la imagen desde otra posición. Esto se debe a que dependiendo de las condiciones de luminosidad, enfoque de la cámara, distancia al objeto y demás factores el color de la figura puede verse con más o menos pureza. Junto a esto se ha observado que es necesario imponer un valor mínimo para que no se confunda la figura con sombras, reflejos y otros elementos de la imagen. Este valor mínimo se ha puesto en 40 ya que se ha comprobado que por debajo de ese valor se almacenaban estructuras con los elementos indicados anteriormente.
- **Luminosidad:** Para un correcto funcionamiento del software ha sido necesario limitar este atributo para despreciar los colores blancos y negros de la imagen. Este se ha debido a que estos colores pueden representarse con cualquier valor de tonalidad interfiriendo en cualquier búsqueda de color. Los umbrales utilizados han sido de 70 para el mínimo y 240 para el máximo.

Capítulo 4

Experimentación

En este capítulo se realiza una descripción de las diferentes pruebas que han sido realizadas sobre el sistema desarrollo en este trabajo con el fin de comprobar su alcance y los factores que influyen en su ejecución. El objetivo final de la experimentación es comprobar que el funcionamiento es el correcto y el grado con el que se han alcanzado las metas propuestas. Para ello se han creado diferentes experimentos para valorar el grado de éxito de cada una de las partes. Debido a que hay partes que no pueden ser separadas de una ejecución completa del sistema se ha optado por crear experimentos con los que se valorarán un conjunto de procedimientos. En cada experimento se pretende centrar la información extraída de uno o varios de los procedimientos que se describen a continuación. Los procedimientos identificados son:

- **Recogida de objetos:** Se pretende comprobar como de efectiva es la recogida de los objetos una vez posicionado a la distancia necesaria. Para ello se pretende probar en que situaciones podrá ser recogido satisfactoriamente el objeto y que factores pueden provocar un fallo en este proceso.
- **Cálculo de la distancia:** Se comprueba como de efectivo es el cálculo de la distancia con el algoritmo desarrollado. Se verá el error cometido en el cálculo de la distancia a los objetos, también se comprobará que interviene en ese error y qué consecuencias puede producir.

- **Reconocimiento de objetos:** Se necesita conocer como de fiable es el reconocimiento de objetos y en que situaciones se puede producir un fallo en esta etapa.
- **Acercamiento al objeto a recoger:** Se comprueba la exactitud del acercamiento a los objetos para realizar su recogida, valorando como interfiere un error de esta fase en el objetivo final del sistema.
- **Búsqueda de objeto:** Con esto se pretende comprobar la eficacia del sistema al determinar si el objeto que se busca se encuentra o no en la imagen inicial y realizar un comportamiento de búsqueda para localizarlo en el entorno cercano. Junto a esto ello se pretende comprobar que el sistema no confunda los elementos del entorno con la figura correcta.

Cada experimento se ha repetido un número de veces, acorde con el procedimiento que se está valorando, comprobando así la consistencia del sistema. A continuación se muestra la descripción de cada uno de los experimentos, los resultados obtenidos, los aciertos y los fallos en las ejecuciones y las conclusiones que se extraen de los datos obtenidos.

4.1. Experimento 1

4.1.1. Descripción del experimento

Para ello se comenzará la ejecución del sistema, en primer lugar se lleva a cabo el proceso de captación de la figura a recoger. Este proceso está definido con mayor detalle en el anexo B.3. En este experimento se realizarán pruebas de recogida con dos objetos diferentes. Esos objetos que se muestran en la imagen 4.1 son un rodillo amarillo y una pelota roja.

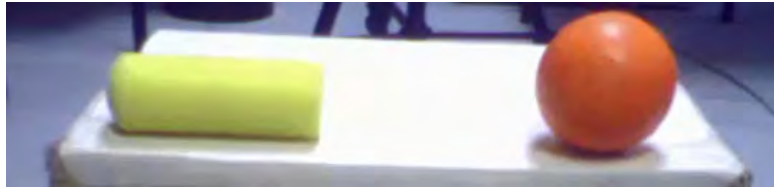


Figura 4.1: Objetos recogidos en el experimento 1.

Acto seguido de captar la figura a recoger se colocará el escenario, para ello se colocará al robot a una distancia de aproximadamente 20 cm de la mesa situando encima de la misma el objeto introducido para su búsqueda en uno de los lados y un objeto de un color diferente en la posición opuesta. En la imagen 4.2 se observa gráficamente la disposición del escenario para su ejecución una vez captada la figura.

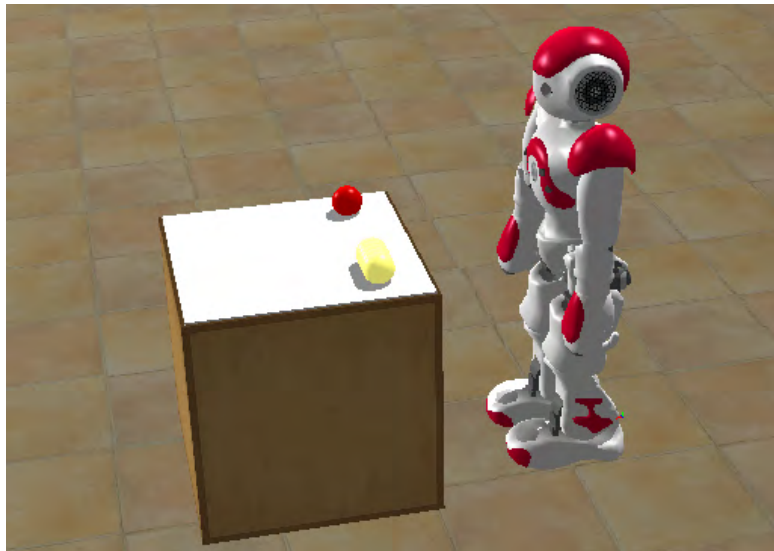


Figura 4.2: Representación experimento 1.

Este experimento se ha ejecutado un total de 5 veces, 3 de ellas introduciendo el cono amarillo para la búsqueda y 2 introduciendo la pelota roja. En el siguiente enlace puede observarse el desarrollo de las pruebas realizadas:

<http://www.veoh.com/watch/v83184086hzCzmGbA>

4.1.2. Resultados obtenidos

Los resultados del experimento han sido satisfactorios. El sistema se ha ejecutado un total de cinco veces, tres de las ejecuciones se han realizado para recoger el rodillo amarillo y dos de ellas para recoger una pelota roja.

En el caso del rodillo amarillo se ha recogido bien en dos ocasiones y ha fallado una vez al situarse el robot a una distancia demasiado cercana del objeto. Esto ha provocado que el comportamiento de recogida no consiguiera sujetar el objeto al existir un error en la posición del robot con respecto al objeto.

Para la pelota roja no ha sido posible su recogida en ninguno de los dos casos debido a que la forma de la figura hace complicado que el robot pueda sujetarlo. Por otra parte la posición del robot con respecto al objeto es la correcta, por lo tanto el fallo de la recogida de este objeto es producido por la forma en sí del propio objeto.

4.1.3. Datos extraídos

A continuación se muestran los problemas y datos extraídos en el experimento sobre la captación de la figura, reconocimiento y recogida de la misma:

- Se ha observado en los experimentos que el tiempo necesario para localizar un objeto que está presente en el entorno cercano es bajo, debido a la necesidad de realizar sucesivas interacciones para realizar el proceso de cálculo de la distancia el tiempo final es elevado. Por ello el tiempo en el proceso de reconocimiento es un punto conflictivo para la duración total del sistema al ser un proceso muy repetido.
- Puede darse el caso de que el objeto se mueva de la posición mientras está ejecutando el sistema debido a golpes del robot sobre la mesa o de la acción directa de la mano en el objeto. Al moverse el objeto mientras esta el proceso de recogida en marcha puede producirse un fallo en la ejecución de este

proceso.

- Se ha comprobado en este experimento la dificultad que tiene el robot de recoger ciertos objetos. Esto es debido a que la articulación de la mano y el tamaño de la misma hacen que resulte difícil sujetar correctamente un objeto. Junto a esto, se ha comprobado que la posición del robot respecto al objeto debe ser muy precisa para poder recogerlo satisfactoriamente. En las pruebas realizadas solo se ha conseguido recoger un pequeño rodillo de color amarillo. Gracias a la longitud y al tamaño del rodillo resulta el elemento más fácil de recoger por el robot con el procedimiento desarrollado.

4.2. Experimento 2

4.2.1. Descripción del experimento

En este experimento se ejecutará el sistema para mostrar y evaluar el cálculo de la distancia al objeto y el posterior acercamiento al mismo.

Para este experimento se colocará el entorno de forma similar al experimento 1. En primer lugar se realizará el proceso de captación de la figura. Para estos experimentos debido a que no pretendemos tomar datos de este proceso se han realizado las ejecuciones con un código modificado en el que se ha introducido el color de la figura mostrada en la imagen 4.3.



Figura 4.3: Cono amarillo usado como figura a recoger.

En primer lugar se situará la mesa con el objeto situado en el centro de la misma y lo más próximo al robot NAO. Se colocará el robot a una distancia de 1

m a la mesa realizando así el proceso de acercamiento al objeto desde una posición más alejada que el experimento 1. Las disposiciones de este experimento pueden verse en la imagen 4.4

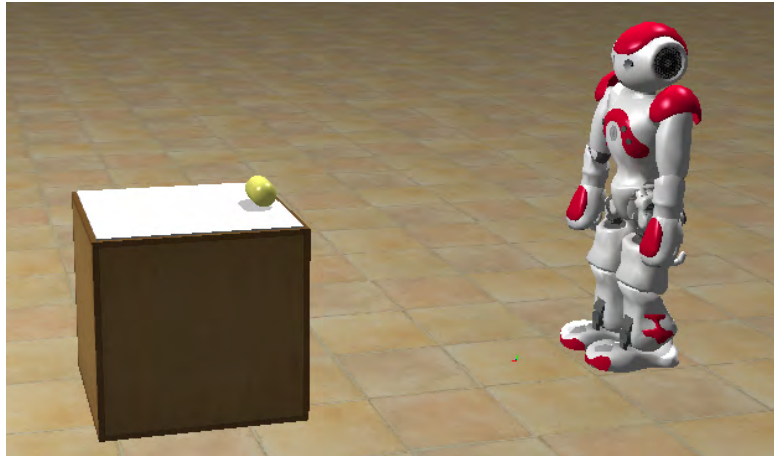


Figura 4.4: Representación experimento 2.

Este experimento se ha ejecutado un total de 5 veces con una disposición del entorno idéntica. En el siguiente enlace puede observarse el desarrollo de las pruebas realizadas:

<http://www.veoh.com/watch/v83486844m2Y2KfBG>

4.2.2. Resultados obtenidos

Los resultados del experimento han sido satisfactorios. Se han observado un total de cuatro aciertos y un fallo en la ejecución del experimento. Este fallo se ha debido a una posición demasiado cercana del robot con respecto a la figura. A continuación se muestran que factores intervienen en este fallo de la ejecución.

4.2.3. Datos extraídos

A continuación se muestran los problemas y datos extraídos en el experimento sobre el cálculo de la distancia y el acercamiento a la figura:

- Sobre el cálculo de la distancia se ha comprobado las distancias en el eje X e Y calculadas por el sistema y se han comparado con las distancias reales. En este experimento se ha podido observar que el cálculo de la distancia no es totalmente preciso y existe una variación mayor cuanto mayor es la distancia. En cambio para el cálculo de distancias pequeñas el proceso funciona con un porcentaje de error muy bajo.
- En el acercamiento se ha observado algunos errores en el movimiento del robot NAO. Debido a la precisión necesaria para el proceso de recogida de los objetos esta imperfección en el movimiento da como resultado una clara dificultad para llevar a cabo el cometido de este sistema.

4.3. Experimento 3

4.3.1. Descripción del experimento

En este experimento se comprobará la búsqueda de los objetos en el entorno. Para ello al comenzar la búsqueda de la figura captada la misma no debe estar contenida en la imagen inicial.

En este experimento se ha confeccionado un entorno similar al del experimento 2. Para ello al igual que en el anterior apartado se ha utilizado un código modificado con el color de la figura a recoger dejando de lado la fase de captación de los objetos. También, de igual forma que en el experimento 2 se ha situado el objeto en el centro de la mesa y en el borde más cercano al robot NAO. En este experimento se ha colocado el robot a una distancia de 1 m al objeto y posicionado en dirección contraria, tal y como se muestra en la imagen 4.5.

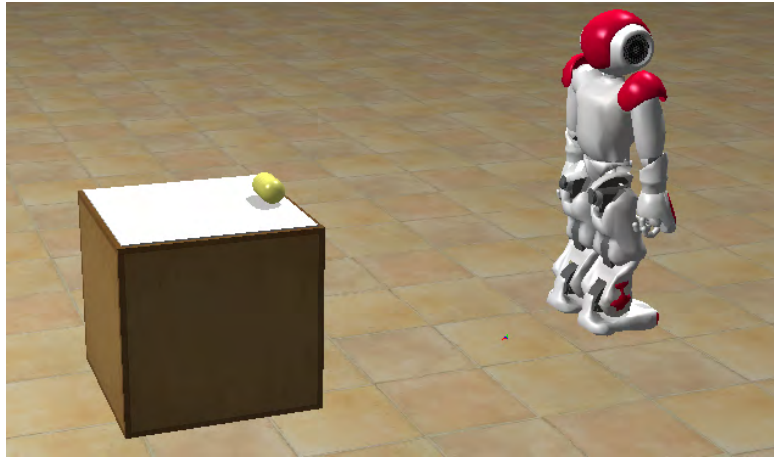


Figura 4.5: Representación experimento 3.

Este experimento se ha ejecutado 3 veces con la misma disposición del entorno. En el siguiente enlace puede observarse el desarrollo de las pruebas realizadas:

<http://www.veoh.com/watch/v83492749AGfgg2ET>

4.3.2. Resultados obtenidos

Los resultados del experimento han sido satisfactorios. Las tres ejecuciones realizadas terminan con acierto en la recogida de los objetos.

4.3.3. Datos extraídos

En este experimento se ha observado lo siguiente:

- A pesar de que en este experimento se ha conseguido recoger los objetos correctamente en todos los casos también se ha observado la presencia de errores en el proceso de búsqueda. Esto es debido a que al ampliar el espacio de búsqueda y procesar los colores presentes en él es posible encontrar atributos de color parecidos y dar como resultado un falso positivo. Antes de comenzar este experimento se realizó una prueba para comprobar que el

funcionamiento del sistema fuera el correcto. En esta prueba se observó que el robot captaba una caja que estaba presente en el entorno y la identificaba como la mesa sobre la que se situarían los objetos. Como medida se retiró del entorno todos los elementos que el sistema pudiera confundir en el proceso de búsqueda. Debido a que en el entorno se pueden encontrar figuras con características de color idénticas a la que se desea buscar no existe una forma de que no se produzcan falsos positivos en el sistema.

- En este se ha observado que el tiempo necesario para llevar a cabo una ejecución completa es relativamente alto. Esto se debe a que el algoritmo de cálculo de la distancia necesita que el objeto captado esté en el centro de la imagen. Por ello es necesario realizar diversos movimientos con los motores del cuello del robot hasta tener la figura completamente centrada. Debido a esto en cada movimiento del cuello del robot precisa de una búsqueda en la nueva imagen tomada con el fin conocer la nueva posición del objeto. Este proceso reiterativo produce un claro aumento en el tiempo de ejecución del sistema.
- Como último dato a destacar de este experimento se ha observado que la incorporación de una búsqueda inicial del objeto no interfiere en el proceso de localización y recogida de la misma.

4.4. Experimento 4

4.4.1. Descripción del experimento

En este experimento se comprobará la localización y recogida de los objetos desde un ángulo diferente a los realizados en los anteriores experimentos.

Este experimento se ha realizado en primer lugar con el código modificado para recoger el cono amarillo. Después de obtener los resultados se ha realizado

otra batería de pruebas para recoger la pelota roja. Para estas pruebas con la pelota roja se ha utilizado la captación de objetos colocando la pelota roja encima de la mesa y el robot a una distancia de 50 cm de la misma.

Una vez captado el objeto se ha dispuesto el escenario con el fin de realizar los experimentos. Para ello se ha desplazado la posición del robot 50 cm a la derecha con respecto al experimento 2. El robot está situado a 1 m de la mesa en el eje X y 50 cm en el eje Y. La figura se ha posicionado en el extremo más cercano al robot de la caja y paralela el borde de la misma. El robot se ha colocado perpendicularmente al objeto. Por lo tanto al acercarse el robot en línea recta a la mesa, este se quedaría oblicuo a la mesa y al objeto. En la imagen 4.6 se puede ver la disposición del escenario para este experimento.



Figura 4.6: Representación experimento 4.

Para este experimento se han lanzado un total de 5 ejecuciones. 3 ejecuciones se han ejecutado para recoger el rodillo amarillo y 2 ejecuciones para recoger la pelota roja. En el siguiente enlace puede observarse el desarrollo de las pruebas realizadas:

<http://www.veoh.com/watch/v83492749AGfgg2ET>

4.4.2. Resultados obtenidos

Este experimento ha dado como resultado fallos en la recogida de los objetos en todas sus ejecuciones. Como se puede observar en el video en las ejecuciones de este experimento el robot no es capaz de recoger el objeto introducido inicialmente por diversos motivos. A continuación se mostrarán los detalles de los fallos producidos.

4.4.3. Datos extraídos

En este experimento se ha observado lo siguiente:

- Posición de recogida incorrecta. En los resultados obtenidos se ha observado que la posición de recogida no es precisa con respecto al objeto. Este error puede deberse a que la orientación del robot con respecto al objeto no es perpendicular como sucede en el resto de experimentos. Esta disposición del escenario debe provocar por tanto que la posición captada por el robot esté ligeramente desplazada. Debido a esto el proceso de recogida de objetos es susceptible de fallar debido a la gran precisión necesaria para llevarlo a cabo.
- Junto a esto también se han observado los límites de la recogida de los objetos. En el caso del cono amarillo la mayor precisión de recogida del objeto se consigue cuando este está totalmente perpendicular al robot, esto se debe a que desde esa posición se logra sujetar más firmemente el objeto. En este experimento se ha observado que al intentar sujetar el objeto con una cierta inclinación hay claras dificultades para recogerlo satisfactoriamente.
- Como ya se ha visto anteriormente el cálculo de la distancia comete mayor fallo cuanto mayor es la distancia a la figura. En las pruebas realizadas, después de realizar el acercamiento del robot a la mesa, se ha observado

que la distancia al objeto es mayor que en los demás experimentos. Por ello, debido a que la distancia al objeto es mayor, es muy difícil alcanzar la posición necesaria para la recogida del mismo.

4.5. Experimento 5

4.5.1. Descripción del experimento

Experimento completo. Ejecución completa del sistema utilizando todas las herramientas contenidas en el mismo.

En este experimento se realizará una ejecución donde poder comprobar todas las variables de nuestro sistema. Para ello realizaremos una ejecución que contenga todas las pruebas realizadas en los anteriores experimentos. En primer lugar se realizará el proceso de introducción de figura. Para ello se colocará el robot dirigido a la mesa y el objeto a captar encima de la misma como se muestra en la imagen 4.7.

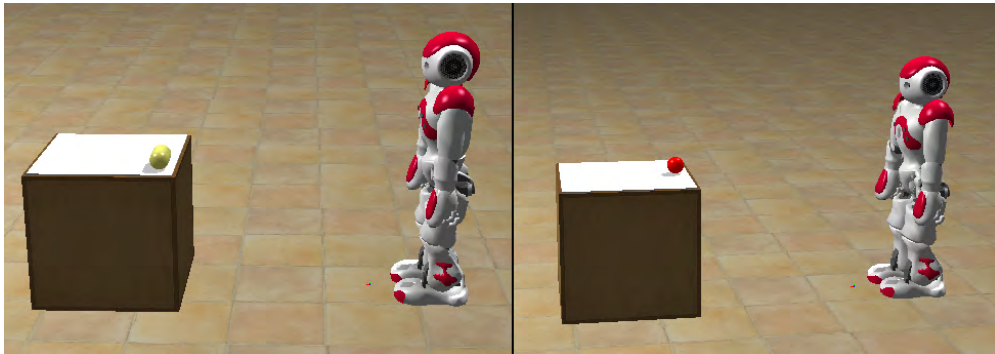


Figura 4.7: Captación figuras experimento 5.

Después de introducir la figura se colocará el entorno antes de comenzar la recogida del objeto. Se situará el objeto a recoger sobre la mesa en uno de los laterales y colocándolo de la forma más cercana al robot. Se colocará un objeto con características de color diferentes a la captada en el lado opuesto. Seguidamente

se situará el robot a una distancia de 1 m de distancia y orientados 90 grados con respecto a la mesa. La disposición final del entorno se puede ver en la imagen 4.8.

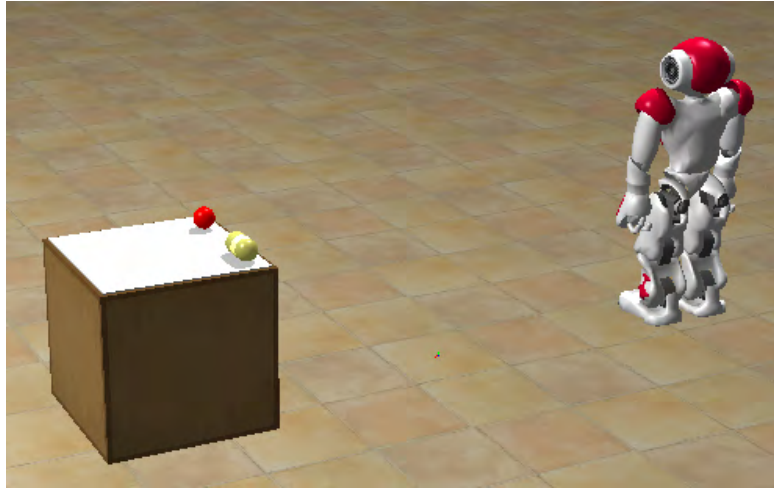


Figura 4.8: Representación experimento 5.

Para este experimento se han lanzado un total de 5 ejecuciones. 3 ejecuciones se han ejecutado para recoger el rodillo amarillo y 2 ejecuciones para recoger la pelota roja. En el siguiente enlace puede observarse el desarrollo de las pruebas realizadas:

<http://www.veoh.com/watch/v83620590bg6J7wrx>

4.5.2. Resultados obtenidos

Los resultados de este experimento han sido satisfactorios. Como se ha visto en los experimentos anteriores se han producido fallos al intentar recoger la pelota roja. En las pruebas realizadas se han recogido correctamente en dos de los tres casos y dando un fallo en la recogida en uno de ellos. Este fallo se ha debido al posicionarse el robot a una distancia errónea con respecto al objeto.

4.5.3. Datos extraídos

En este experimento se ha observado lo siguiente:

- En este experimento final se ha extraído información del completo funcionamiento de sistema. Con estas pruebas realizadas se muestra por tanto el correcto funcionamiento de todas las partes tanto por separado como en conjunto, así como el alcance de cada una de ellas.
- Como último dato reseñable se ha comprobado la funcionalidad del sistema modificando el entorno mientras se está realizando la búsqueda de un objeto en el mismo. Como resultado se ha visto que el sistema es capaz de procesar estos cambios en el entorno y actuar consecuentemente para realizar la recogida del objeto. Estos cambios en el entornos solo son perceptibles por el sistema mientras el proceso de recogida no se haya activado.

4.6. Análisis de la experimentación

Mediante los experimentos realizados se ha extraído información del sistema fase por fase describiendo el alcance y límite de cada experimento realizado. A continuación se intentara dar una visión general de la información extraída en el conjunto de los experimentos:

- **Límites en el proceso de recogida.** Se han comprobado en que supuestos el sistema es capaz de realizar una recogida correcta de la figura y las causas por las que se puede dar lugar a un fallo. En estos experimentos se ha comprobado la dificultad existente para realizar la recogida de los objetos según su forma y su tamaño. De ello se ha extraído que tanto la forma como el tamaño son cualidades fundamentales para una ejecución correcta. De las pruebas realizadas se extraído que el objeto a recoger debe quedarse completamente sujeto en la mano del robot cuando este realiza el cierre de la misma. Si esto no se produce el objeto tendera a caerse en el transcurso de la recogida. Se ha visto también que en el caso del cono amarillo el robot

debe quedarse perpendicularmente al objeto, en caso contrario la sujeción del objeto por el robot no se realizará de forma correcta y por lo tanto se producirá un fallo en la recogida. Por ultimo tener en cuenta que para que el proceso de recogida se realice correctamente el robot debe quedarse en la posición adecuada para ello. Esta posición debe ser de gran precisión tolerando únicamente un error de pocos centímetros.

- **Calculo de distancia y posicionamiento.** Ya se ha comprobado anteriormente que la posición de recogida de una figura debe ser lo más exacta posible para realizar la recogida correctamente. Por ello se han realizado pruebas para comprobar cómo influye el cálculo de la distancia y el movimiento del robot en la recogida de objetos. En primer lugar se han realizado comprobaciones del cálculo de la distancia con respecto a la distancia real a la figura. En estas comprobaciones se ha obtenido que al realizar el cálculo de la distancia esta tiene un ligero error con respecto a la distancia real. Con respecto a ello se ha comprobado que el error cometido es mayor cuanto mayor es la distancia al objeto, obteniendo un error muy bajo en el cálculo de distancias pequeñas. Junto a esto cabe destacar que, sucede un hecho parecido al realizarse los movimientos del robot. Al enviar una petición para andar una distancia determinada se obtiene que el proceso de movimiento comete un ligero error. Debido a las imperfecciones de estos dos procesos se obtiene como resultado una variación de la posición entre la que debe colocarse el robot para la recogida y la posición final. Esta imperfección del sistema puede dar lugar a que no se realice la recogida del objeto correctamente y por lo tanto desemboca en un fallo del sistema.
- **Reconocimientos de figuras.** El proceso de reconocimiento de figuras ha dado un gran resultado en el sistema consiguiendo una ausencia casi total de errores en este proceso. Debido a la importancia que se le ha dado a este proceso se ha intentado que la figura identificada tenga unos atributos de color casi idénticos a la figura buscada para evitar que se realice una identi-

ficación errónea de la figura. Bien es cierto que debido a que la búsqueda de las figuras se realiza únicamente por las cualidades de color resulta posible que el sistema identifique en el entorno una figura con cualidades de color parecidas, e incluso idénticas.

- **Búsquedas de figuras.** Se ha comprobado el proceso de búsqueda de las figuras en el entorno. Como se ha visto en los experimentos se ha realizado un desplazamiento del robot para aumentar el espacio de visión del mismo. Este proceso ha sido aplicado sin problemas consiguiendo identificar la figura deseada sin estar directamente en el campo de visión del robot. Como resultado se obtiene que el sistema sea capaz de localizar un objeto mediante una búsqueda de 360 grados en el entorno y a una distancia menor de 2 m.

Capítulo 5

Gestión del trabajo

En este capítulo se realiza una descripción detallada del ciclo de vida que se ha utilizado para la realización de este trabajo. Además se realiza una estimación del tiempo y del coste necesario para realizarlo.

5.1. Descripción de las fases del trabajo

Para llevar a cabo la planificación se dividió el proceso de desarrollo en diferentes fases:

- **Análisis:** Durante esta fase se ha realizado un análisis del problema que se quiere resolver mediante este trabajo. El análisis del sistema es una de las tareas más importantes en el ciclo de vida del desarrollo de software, ya que de ellos depende el correcto funcionamiento del nuevo sistema. En esta fase se han identificado y documentado todas las partes del sistema.
- **Diseño:** El objetivo de esta fase ha sido la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

- **Implementación:** En esta fase se ha realizado la implementación total del sistema diseñado en la fase anterior. Para ello se ha dividido la implementación en los tres bloques identificados en el apartado *3.3 Diseño del sistema*.
- **Pruebas:** En esta fase se han realizado una serie de pruebas para comprobar el correcto funcionamiento del sistema. Esta fase se realizó para cada bloque desarrollado en la fase de implementación, comprobando cada conjunto de funcionalidades por separado.
- **Evaluación:** Tras obtener una primera versión funcional del sistema se ha comprobado que el funcionamiento sea el adecuado y que se cumplan los objetivos definidos inicialmente.
- **Documentación:** Esta fase consiste en la creación de este documento.

5.2. Planificación

En la tabla 5.1 se presentan las diferentes fases que comprenden el ciclo de vida utilizado para el desarrollo de este trabajo. Cada una de estas fases se encuentra dividida en tareas más pequeñas que describen las diferentes actividades que han sido desarrolladas en este trabajo. Para cada una de las actividades identificadas se presentan las fechas estimadas, así como las fechas reales que se han invertido. Debido a que las horas diarias reales se han repartido desigualmente durante la realización del trabajo se van a mostrar en una columna conjuntamente con cada tarea realizada.

Como se puede ver en la tabla 5.1 ha habido una desviación notable en las fechas estimadas en el trabajo. Esta desviación se ha debido tanto a diversos problemas encontrados durante la realización del trabajo descritos en el apartado 6.3

A continuación se presentan dos diagramas de Gantt (Figura 5.1 y Figura 5.2). El primero presenta las horas estimadas de cada una de las tareas y el segundo

Estimacion				Real		
Tarea	Comienzo	Fin	Horas	Comienzo	Fin	Horas
Análisis	05/08/13	01/09/13	112 h	05/08/13	15/09/13	168 h
Investigación trabajos anteriores	05/08/13	08/08/13	16 h	05/08/13	08/08/13	16 h
Estudio de C++	09/08/13	17/08/13	36 h	09/08/13	26/08/13	72 h
Estudio de visión artificial	18/08/13	27/08/13	40 h	27/08/13	04/09/13	40 h
Estudio del Framework de NaoQi	28/08/13	01/09/13	20 h	05/09/13	15/09/13	40 h
Diseño	02/09/13	15/09/13	56 h	16/09/13	29/09/13	56 h
Diseño de la arquitectura	02/09/13	15/09/13	56 h	16/09/13	29/09/13	56 h
Implementación y pruebas	16/09/13	15/12/13	328 h	30/09/13	22/03/14	382 h
Implementación capa color v0.1	16/09/13	20/09/13	20 h	30/09/13	02/10/13	16 h
Pruebas bloque capa color v0.1	21/09/13	21/09/13	4 h	03/10/13	04/10/13	8 h
Implementación capa figura v0.1	22/09/13	06/10/13	60 h	05/10/13	15/10/13	44 h
Pruebas capa figura v0.1	07/10/13	08/10/13	8 h	16/10/13	20/10/13	20 h
Implementación capa color v0.2	09/10/13	11/10/13	12 h	21/10/13	08/11/13	46 h
Pruebas bloque capa color v0.2	12/10/13	12/10/13	4 h	09/11/13	10/11/13	9 h
Implementación capa figura v0.2	13/10/13	17/10/13	20 h	11/11/13	07/12/13	77 h
Pruebas capa figura v0.2	18/10/13	19/10/13	8 h	08/12/13	12/12/13	16 h
Implementación capa robot v0.1	20/10/13	08/11/13	80 h	13/12/13	06/01/14	63 h
Pruebas capa robot v0.1	09/11/13	13/11/13	20 h	07/01/14	10/01/14	16 h
Implementación capa robot v0.2	14/11/13	23/11/13	40 h	11/01/14	27/01/14	26 h
Pruebas capa robot v0.2	24/11/13	26/11/13	12 h	28/01/14	29/01/14	8 h
Implementación sistema final	27/11/13	06/12/13	40 h	30/01/14	22/03/14	33 h
Evaluación	07/12/13	16/12/13	40 h	23/03/14	18/04/14	29 h
Evaluación del sistema	07/12/13	16/12/13	40 h	23/03/14	18/04/14	29 h
Documentación	17/12/13	28/01/14	80 h	19/04/14	15/05/14	65 h
Documentación del trabajo	17/12/13	28/01/14	80 h	19/04/14	15/05/14	65 h

Tabla 5.1: Definición de tiempos y tareas del procesos de desarrollo del software.

con las horas reales dedicadas finalmente.

Por ultimo cabe destacar el rol del jefe de trabajo que ha sido transversal al desarrollo del mismo y con el que se ha contado con una participación de 30 horas.

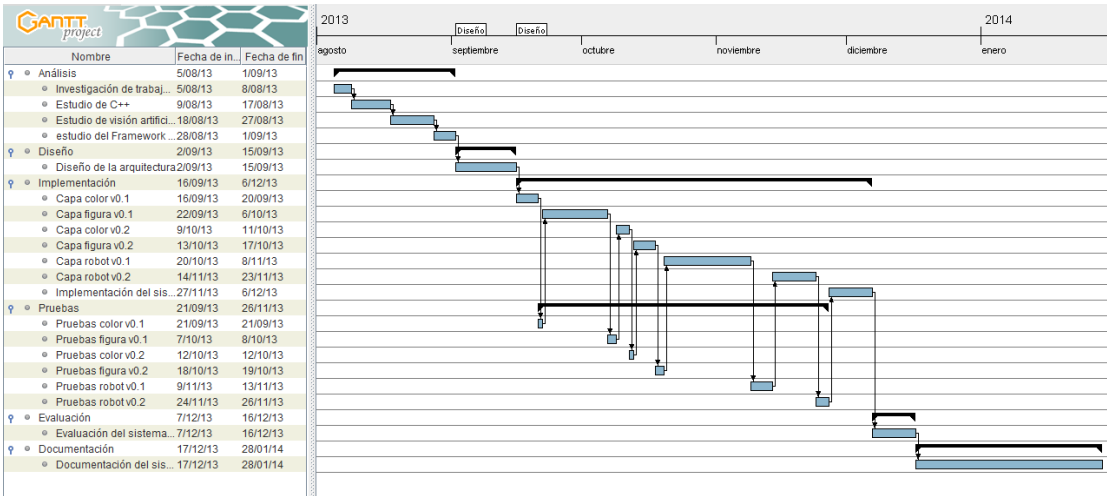


Figura 5.1: Diagrama de Gantt con las horas estimadas.

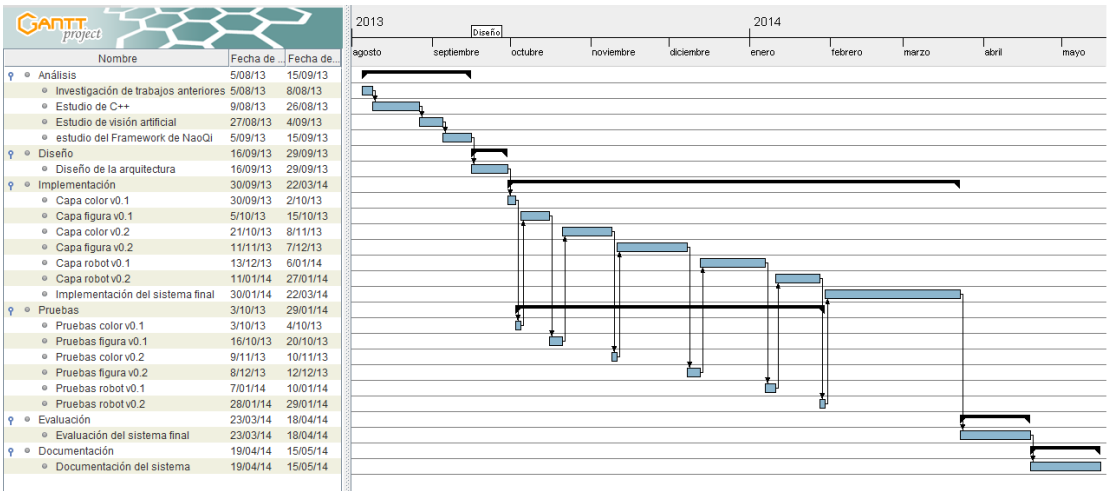


Figura 5.2: Diagrama de Gantt con las horas reales.

5.3. Presupuesto

En esta sección se presentan los costes necesarios para el desarrollo del trabajo. A continuación se van a desglosar los costes producidos entre diferentes tipos.

5.3.1. Recursos humanos

Para la estimación de los salarios se ha tomado como referencia el salario del puesto correspondiente en el mercado actual [21]. En la tabla 5.2 se incluye una tabla con los costes en recursos humanos del trabajo.

Perfil	Salario hora (€)	Horas dedicadas	Total bruto (€)
Analista	8.33	168	1,399.44
Diseñador	13.02	56	729.12
Programador	6.25	382	2,387.50
Responsable de calidad	8.33	29	338.00
Documentalista	5.20	65	241.57
Jefe de trabajo	13.02	30	396.00
Total			5,491.63

Tabla 5.2: Costes salariales personal.

5.3.2. Recursos físicos

En la tabla 5.3 se incluyen los costes de los recursos físicos del trabajo.

Concepto	Precio (€)	Cantidad	Amortización	Meses	Imputable
Ordenador Asus N56V	799.00	1	16.64	9	149.81
Robot Nao	12,000.00	1	250.0	3	750.00
Material de oficina	50.00	-	-	-	50.00
Transporte	200.00	-	-	-	200.00
Total					1,149.81

Tabla 5.3: Costes físicos.

5.3.3. Costes indirectos

Costes indirectos del trabajo. Tabla 5.4.

Concepto	Coste (€/mes)	Meses	Coste imputable
Luz	100.00	9	900.00
Internet	50.00	9	450.00
Total			1,350.00

Tabla 5.4: Costes indirectos.

5.3.4. Beneficios y riesgos

Beneficios y riesgos del trabajo. Tabla 5.5.

Concepto	Porcentaje	Coste total (€)
Coste del trabajo	-	7,991.44
Beneficios	15 %	1,198.72
Riesgos	5 %	399.57
Total		9,589.58

Tabla 5.5: Beneficios y riesgos del trabajo.

5.3.5. Cálculo de coste total

Coste total del trabajo. Tabla 5.6.

Coste del trabajo	IVA (21 %)	Coste total del trabajo
9,589.58€	2,013.81€	11,603.39€

Tabla 5.6: Coste total del trabajo.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones obtenidas tras la realización de este trabajo. En primer lugar se presentan las conclusiones generales obtenidas tras la realización de este trabajo; a continuación se presentan las conclusiones referentes a los objetivos presentados al inicio de este documento; y por último se presentan los problemas encontrados y los trabajos futuros que podrán realizarse a partir del trabajo presentado en este documento.

6.1. Conclusiones generales

Como ya se ha visto en el apartado 4, los resultados obtenidos han sido satisfactorios, consiguiendo cumplir los objetivos marcados. Así mismo se ha cumplido el objetivo global del trabajo con éxito. Aun así el sistema desarrollado tiene un margen de mejora alto pudiendo dar una mayor libertad al proceso de búsqueda. Esto se debe a que el objetivo principal de este trabajo es la recogida en sí de los objetos una vez localizados y no la búsqueda de los mismos. Por esto la fase de búsqueda se ha realizado en segundo plano limitando el proceso a objetos del entorno más cercano y sin tener en cuenta obstáculos. Esto se ha debido a que la funcionalidad de recogida de los objetos tiene como objetivo servir de apoyo a trabajos más amplios en el que se pueda ser acoplado esta funcionalidad sa-

tisfactoriamente. Mientras que la parte de búsqueda es demostrativa pudiendo ser sustituida o acompañada por un algoritmo de localización más adecuado al entorno y proporcionando más libertad.

Por todo lo dicho anteriormente se ha concluido diciendo que el sistema desarrollado es correcto y puede aportar una gran ayuda a futuros trabajos en los que se puedan acoplar estas funcionalidades fácilmente.

6.2. Conclusiones referentes a los objetivos

A continuación se analizarán los objetivos marcados en el apartado 1.3 y se mostrarán las conclusiones que puedan obtenerse de su realización.

- **Estudio previo**

El estudio inicial realizado tanto del framework de NAOqi como de la visión artificial ha sido de gran importancia para realizar la implementación de sistema. Los conocimientos adquiridos en esta fase fueron adecuados para comenzar a trabajar, no obstante el mayor conocimiento de estas áreas se obtuvo al comenzar a trabajar poniendo a prueba lo aprendido.

- **Detección de objetos**

Este apartado supuso un gran trabajo dentro del desarrollo del trabajo debido a las dificultades presentes en la detección de objetos mediante la visión artificial. Se necesitó trabajar en esta parte hasta conseguir encontrar inequívocamente un objeto dentro de un escenario con ruido. Esto era un proceso de gran importancia ya que suponía la base de todo el resto del trabajo, por eso se necesitó realizar una gran cantidad de correcciones hasta conseguir que el funcionamiento fuera el deseado. Finalmente se puede concluir diciendo que se ha cumplido correctamente el objetivo marcado al comienzo del trabajo.

- **Extracción de información del entorno**

La extracción de la información del entorno ha sido un proceso muy delicado necesario para el resto del trabajo. En primer lugar se ha realizado la búsqueda de objetos dentro de la imagen calculando sus propiedades básicas del mismo, tales como, región que ocupan en la imagen y los atributos del color. Esto se realizó sin grandes dificultades y se almacenó para su posterior tratamiento. Cabe destacar que ha sido imposible extraer la información necesaria para el cálculo de la distancia al objeto. Por ello ha sido necesario introducir datos externos para realizar el cálculo haciendo que el sistema sea inflexible si se modifica la altura de la mesa. En general podemos concluir que la realización de este apartado es satisfactoria ya que cumple el objetivo marcado aunque no dispone de una libertad absoluta.

- **Implementación de primitivas motrices para el robot NAO**

La implementación de las primitivas de movimiento para el robot Nao ha sido satisfactorio y se ha realizado fácilmente gracias a las herramientas proporcionadas por el framework de NAOqi y a su conocimiento adquirido de las mismas en las fases anteriores.

- **Agrupación de todos los componentes**

La agrupación de todos los componentes era una fase determinante ya que en este momento se comprobaba que las partes desarrolladas estaban correctamente implementadas para acoplarse entre sí. Esta parte se realizó correctamente y sin problemas dejando el sistema terminado.

- **Evaluación del sistema**

La evaluación ha permitido poner a prueba el sistema utilizándolo en un entorno real y evaluando por separado las diferentes funcionalidades. Los experimentos realizados han sido correctos y se ha mostrado en gran medida el alcance del trabajo dando ideas para desarrollos futuros del mismo.

6.3. Problemas encontrados

En líneas generales se puede decir que el trabajo ha transcurrido sin demasiadas dificultades. Bien es cierto que han surgido problemas durante el transcurso del mismo. Los problemas más destacables han sido:

- **Mal funcionamiento del robot**

El funcionamiento del robot Nao no era el correcto debido a que no se podía acceder a una de las cámaras del mismo. Esto impidió avanzar el trabajo hasta que se arregló el problema. Durante el tiempo que no estuvo operativo el robot se trabajó en partes del trabajo de poca relevancia.

- **Ruido en el reconocimiento de figuras**

El ruido presente en la imagen dificultaba a menudo la correcta ejecución del sistema. Esto se debía a que la figura que se buscaba se confundía con zonas de la imagen con características de color parecidas. Para solventar este error se trabajó en diversas partes del código para aumentar la precisión con la que se localizaban los objetos eliminando el mayor ruido de la imagen posible. Debido a que los resultados obtenidos no eran los esperados se realizó una función para ordenar las regiones de color encontradas respecto al parecido con el objeto a buscar. Finalmente esto solucionó el problema por completo dando una gran precisión a la búsqueda de un objeto en la imagen.

- **Calculo erróneo de la distancia**

Debido a la implementación desarrollada para el cálculo de la distancia a los objetos, los errores presentes en las variables de entrada producen una salida con un alto grado de error, produciendo que este proceso sea incorrecto. Se observó que el cálculo de la distancia no coincidía con la distancia real en las pruebas realizadas en posiciones alejadas. Esto se debía a que la cámara usada para este cálculo tenía una desviación con respecto a la horizontal de aproximadamente 1 grado lo que producía el error en el cálculo de la

distancia. Esto se solucionó introduciendo una variable para compensar esta desviación.

- **Problemas con el simulador**

El simulador no representaba fielmente el comportamiento de un robot NAO real. Por esto se trabajó con el simulador en partes de menor importancia del código y dejando el resto de pruebas para el robot real.

6.4. Trabajos futuros

Las principales mejoras que se podrían aplicar al trabajo en un futuro son:

- **Proceso de búsqueda:** El algoritmo de búsqueda desarrollado en el sistema se trata de un algoritmo muy sencillo. Para posteriores desarrollos se podría implementar un algoritmo de búsqueda más avanzado y acorde con las necesidades abarcando un mayor area de busqueda.
- **Mejorar el acercamiento al objeto a recoger:** El acercamiento realizado actualmente comete errores frecuentemente a la hora de calcular la distancia y de recorrerla. Esto se debe, por una parte a la sensibilidad a errores de la función usada para calcular la distancia, y por otra, a las desviaciones producidas al andar por un funcionamiento inadecuado de algunos de los motores del robot. Debido a eso se ha realizado más de un acercamiento para poder recoger el objeto correctamente. Por esto, este proceso es susceptible de mejora pudiendo disminuir considerablemente el tiempo de ejecución.
- **Aumento de libertad en la recogida de los objetos:** Debido a las características del sistema solo se pueden recoger objetos situados en una mesa de 31cm de alto. Esto se debe tanto a la función utilizada para calcular la distancia como al comportamiento de recogida del objeto. Dando libertad

a ambos procedimientos se podría conseguir crear una recogida de objetos capaz de actuar en entornos con más parecido a la realidad.

- **Comportamiento post recogida:** En este trabajo se ha implementado un comportamiento muy básico por el cual el robot Nao muestra el objeto al usuario después de haberlo recogido. Este comportamiento se puede ampliar considerablemente haciendo que el robot, después de recoger un objeto, lo coloque en una posición determinada según sus atributos. De esta forma se podría conseguir un sistema de ordenación automático de objetos completamente autónomo. Para esto, además de crear un comportamiento final para colocar los objetos, se necesitaría extraer más información de los objetos a partir de la imagen como puede ser, forma, tamaño, etc.

Apéndice A

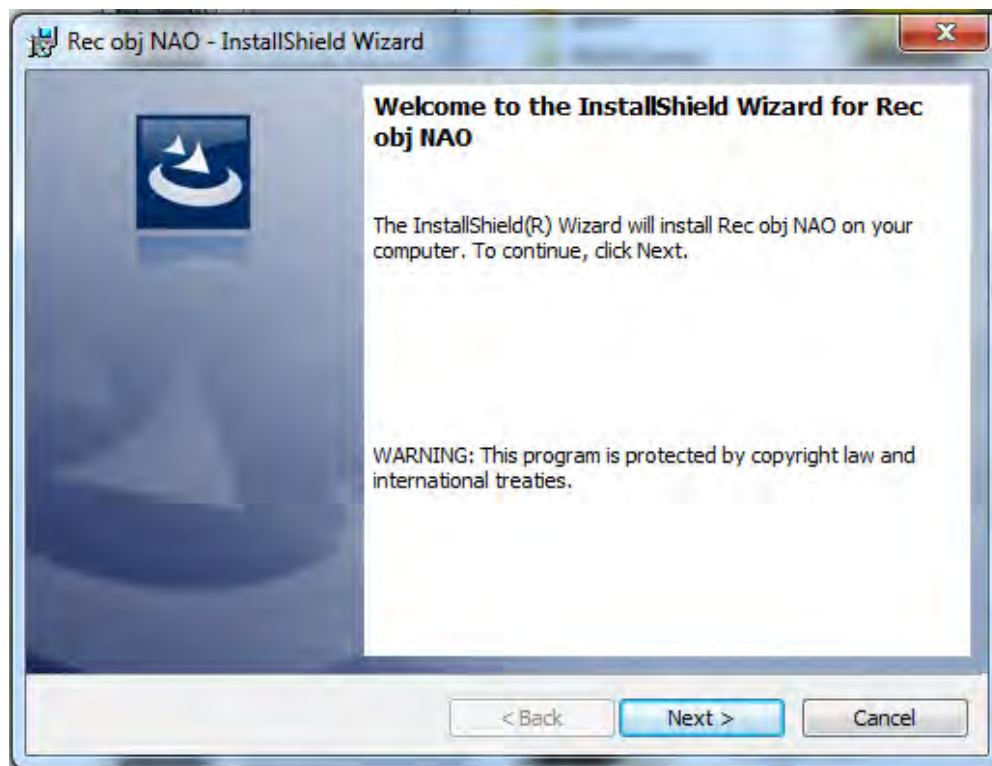
Manual de instalación

A continuación se mostrará como se realiza la instalación del sistema y de algunas heramientas que pueden servir de ayuda:

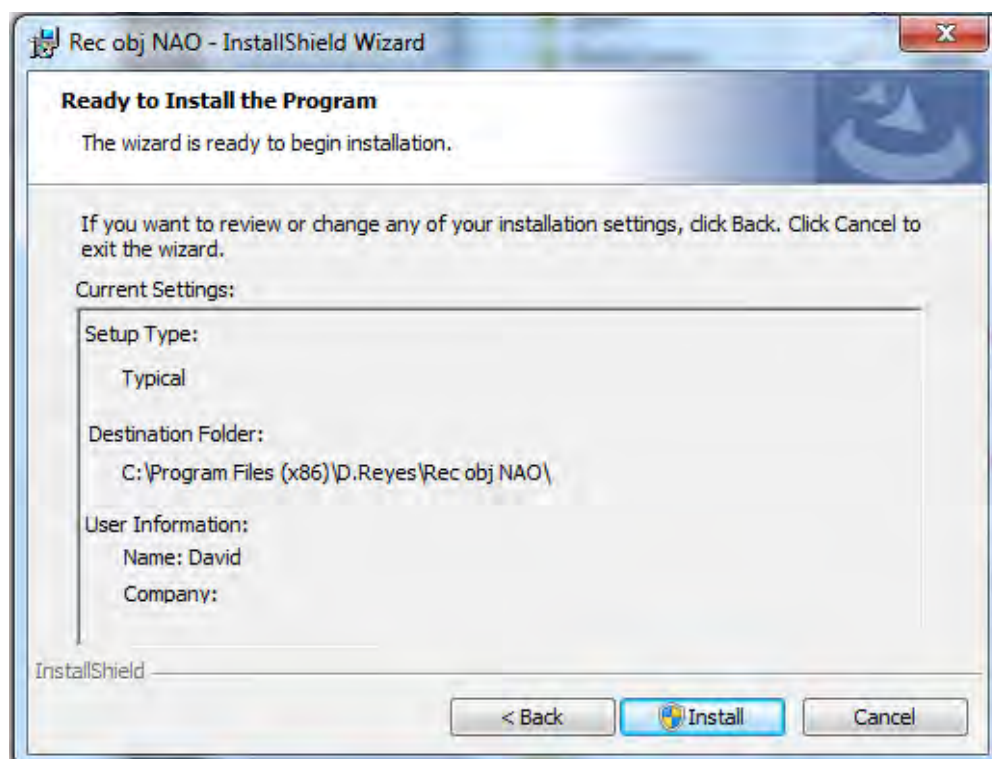
A.1. Instalación del sistema

A continuación se detalla cómo debe realizarse la instalación del sistema para su ejecución.

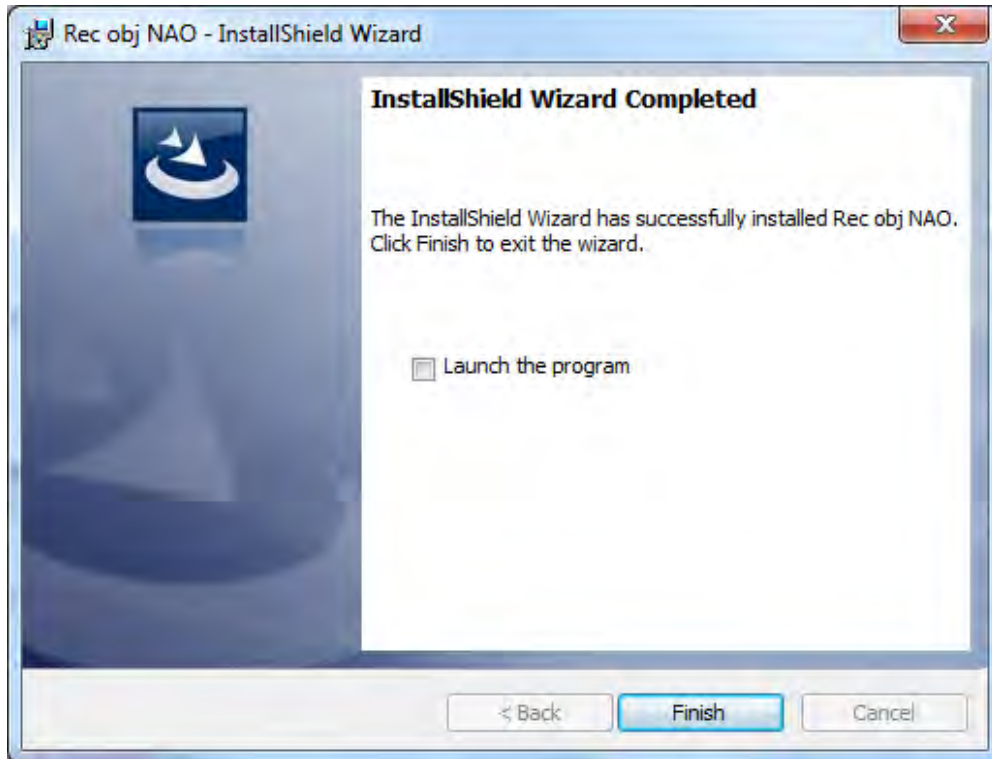
1. Descargarse el fichero ejecutable desde la dirección <https://www.dropbox.com/s/gdoigh3x63ty8id/Buscar%20objetos%20NA0%20-%20Instalador.rar?dl=0>.
2. Descomprimir el archivo descargado.
3. Introducirse en la carpeta extraída y ejecutar el archivo “setup.exe”.
4. Aparecerá una ventana de instalación y haremos clic en “Next”.



5. A continuación aparecerá una ventana y haremos clic en “Install” comenzando el proceso de instalación.



6. Una vez haya terminado el proceso de instalación haremos clic en “Finish” y ya podremos ejecutar el sistema.

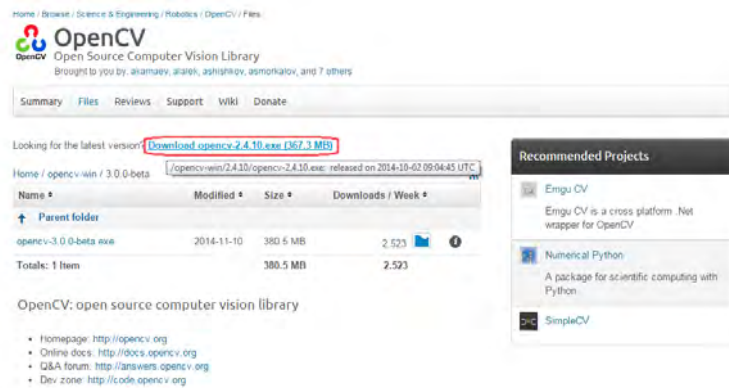


A.2. Instalación de OpenCv

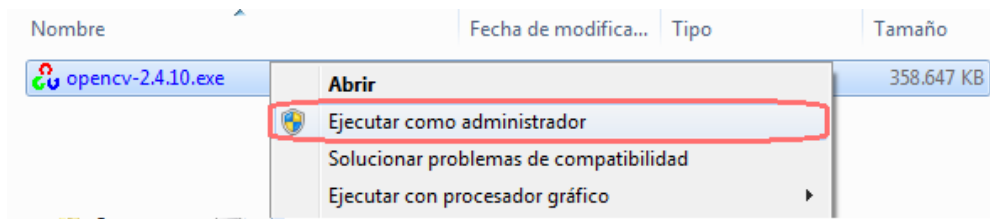
OpenCV (Open Computer Vision) es una librería de código abierto destinada al procesamiento de imágenes, principalmente en tiempo real. Esta librería ha sido utilizada para realizar la detección de objetos del sistema.

En este tutorial se ha usado Windows 7 x64bit y la última versión 2.4.9 de OpenCV. No se garantiza que la instalación sea efectiva para otras distribuciones de Windows ni para otras versiones de la librería.

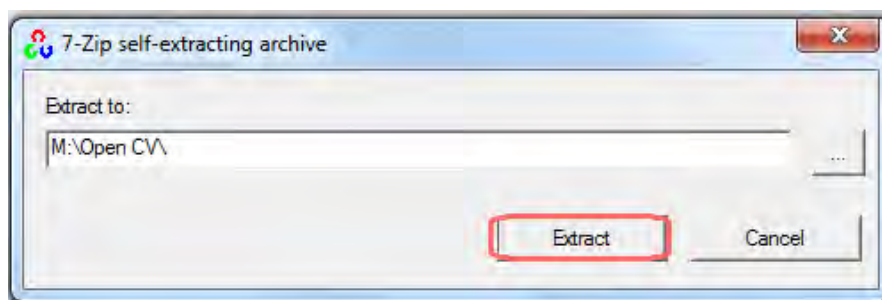
1. Accederemos a la página oficial de OpenCV <http://opencv.org/> desde la que podremos descargar la última versión de la librería para Windows.



2. Una vez descargada la librería pulsaremos el botón derecho del ratón y seleccionaremos la opción *Ejecutar como administrador*.

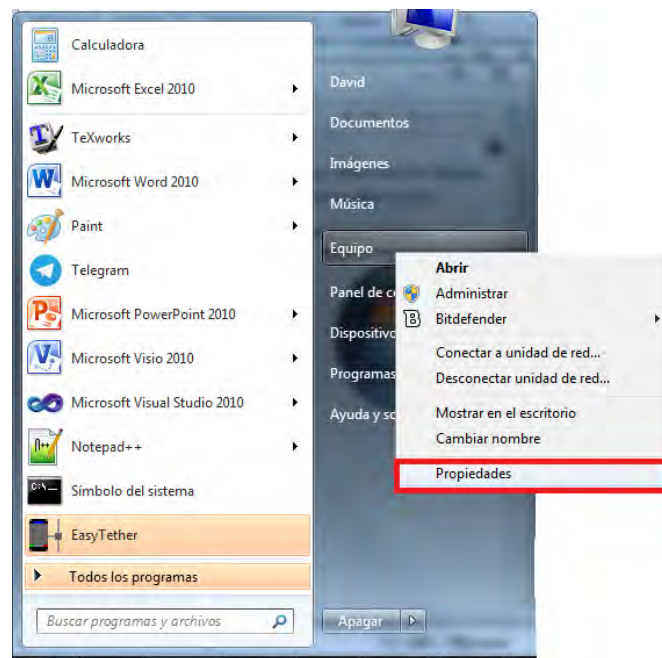


3. Nos aparecerá la ventana donde se seleccionara la ruta donde se extraerán los archivos.

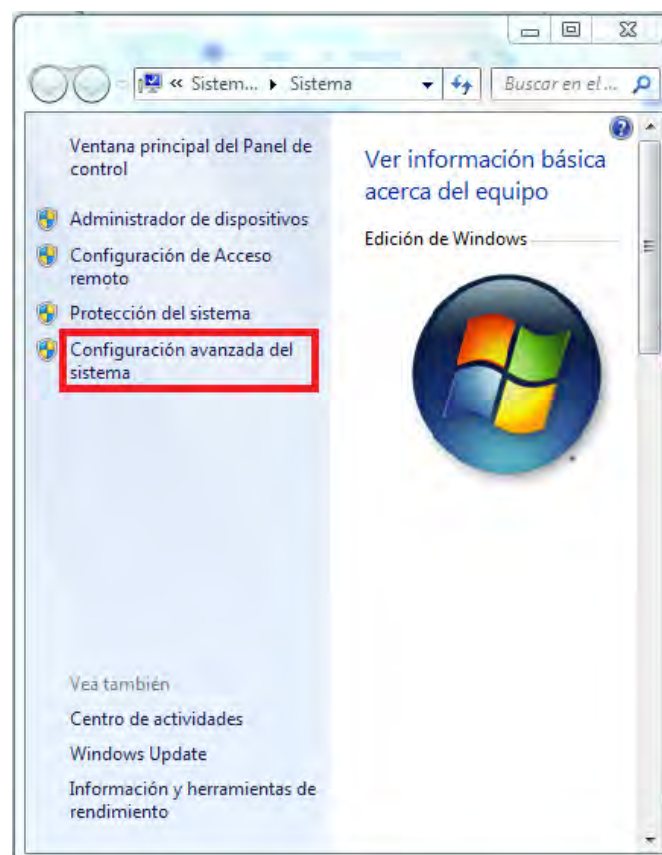


4. Después de que se finalice la extracción se debe incorporar la ruta que contiene los archivos de la biblioteca al Path del sistema:

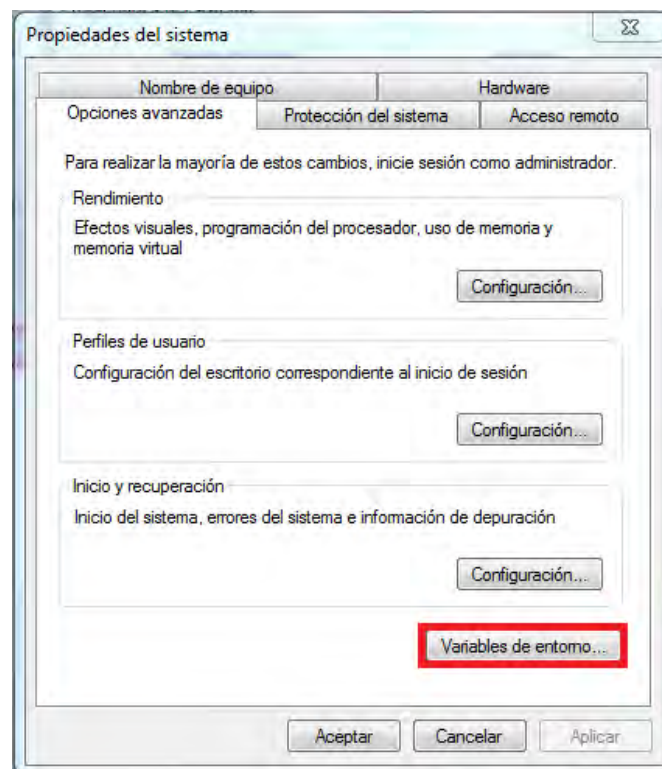
- a) Con el botón derecho del ratón pulsamos sobre *Equipo* y seleccionamos *Propiedades*.



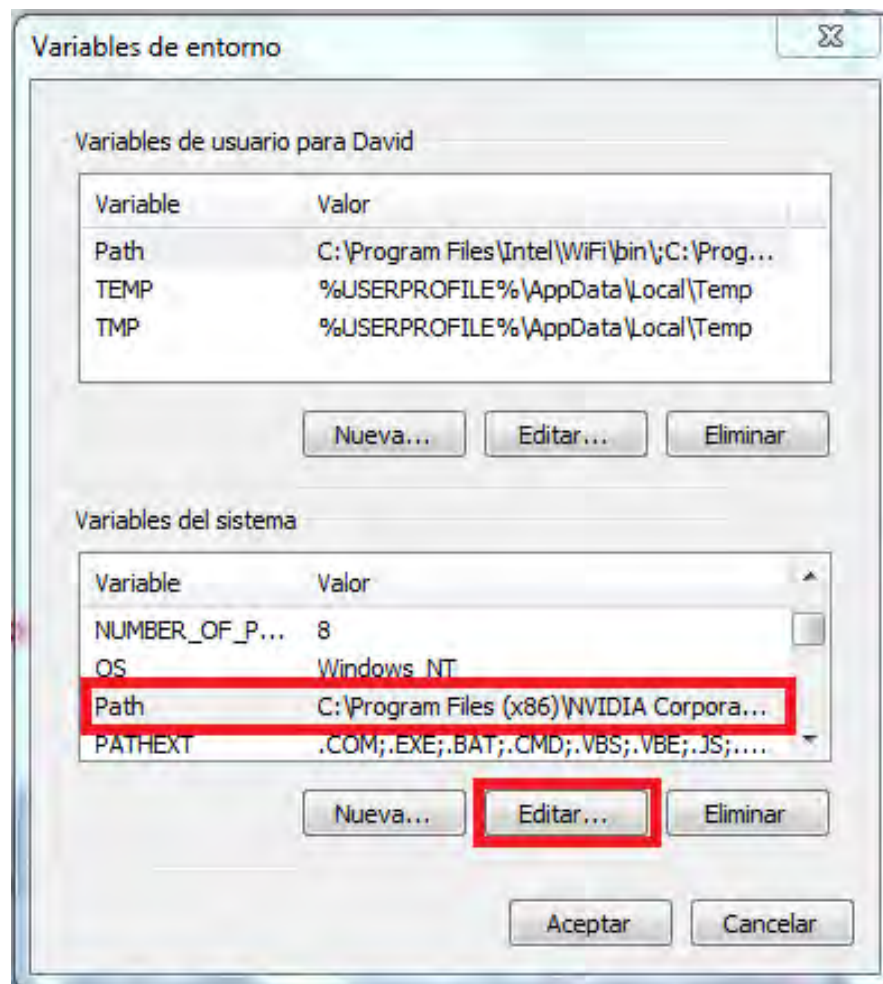
- b) En el menú de la izquierda pulsamos sobre *Configuración avanzada del sistema*.



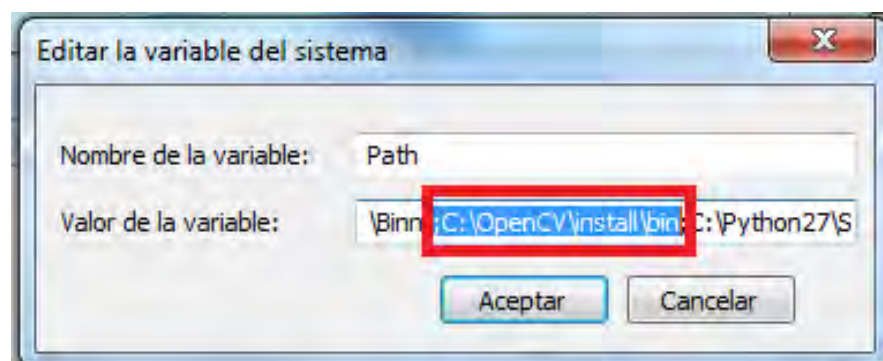
c) En *Opciones avanzadas*, pinchamos sobre *Variables de entorno*.



d) En el apartado de *Variables del sistema*, seleccionamos *Path* y editamos.



- e) En el *Valor de la variable* añadimos al final, con cuidado de no eliminar nada y separado por un punto y coma ';', la ruta de la carpeta.



A.3. Instalación de Choregraphe

Choregraphe es una herramienta proporcionada por Aldebaran Robotics que permite hacer una conexión rápida y fácil con el robot NAO y ejecutar comportamientos ya definidos. La instalación de este software se realiza de la siguiente manera:

1. Descargue el archivo instalador *choregraphe-suite-x.x-win32-setup.exe* desde la dirección <https://community.aldebaran.com/en/resources/software> de la página web de Aldebaran Robotics.
2. Hacer doble clic en el archivo descargado para comenzar la instalación.
3. Aceptar las siguientes ventanas del asistente de instalación.

Apéndice B

Manual de usuario

B.1. Antes de comenzar...

Asegúrese de tener instalada la versión de NAOqi 1.12.3 en el robot NAO y un equipo sobre el que ejecutar el sistema con Windows 7 o superior. Si no dispone de los elementos necesarios para ejecutar el sistema sobre su equipo consulte antes el manual de instalación. Una vez tenga todo listo y disponga de los elementos necesarios puede comenzar a preparar el sistema:

1. Encienda el robot NAO y conéctelo al equipo sobre el que se ejecutará el software.
2. Una vez conectado utilice un programa externo, como Choregraphe, para adoptar la posición inicial sobre el robot NAO y así comprobar la conectividad del mismo con el equipo.

!!!YA ESTA LISTO PARA EJECUTAR EL SISTEMA!!!

B.2. Comience a usar el sistema

Una vez está todo preparado ya puede ejecutar el sistema. Pulse doble clic al archivo ejecutable y listo. Esta conexión se realizará mediante la IP 127.0.0.1

y el puerto 9559. Si desea conectar el robot mediante una ip o puerto diferente ejecute el sistema por consola añadiendo a continuación `-pip [IP] -pport [PUERTO]`”Tenga en cuenta que una vez se ejecute el sistema el robot adoptará la posición inicial, por lo que evite posiciones inadecuadas del robot que puedan provocar daño sobre el mismo al intentar adoptar la posición de partida.

B.3. Introducción del objeto a buscar

Antes de poder comenzar la búsqueda es necesario introducir un objeto a buscar. De este objeto se extraerán los atributos de color usados para la ejecución del sistema. La entrada de este objeto se realizará mediante la intervención del usuario, siendo la única parte en la que es necesaria la presencia del mismo.

Para introducir el objeto en el sistema se siguen los siguientes pasos:

1. Colocar el objeto a buscar sobre la mesa de forma que sea visible desde la posición del robot. El robot tiene que encontrarse a una distancia de 50cm aproximadamente de la mesa.
2. Pulse el botón delantero de la cabeza del robot para comenzar la búsqueda del objeto.
3. El robot tomará una imagen del entorno y la procesará en busca de objetos. Una vez procesada se mostrarán los objetos encontrados al usuario, pulsando el botón delantero de la cabeza del robot si es la figura deseada, o el botón trasero si no lo es.
4. Una vez seleccionado el objeto se preguntará al usuario si desea introducir el objeto que servirá de mesa o si se seleccionan los atributos por defecto. Si se desea introducir se pulsará el botón delantero de la cabeza del robot y se

repetirá el paso anterior, si no se pulsará el botón trasero para seleccionar el valor por defecto.

Este proceso puede verse con gran detalle en el video a través del siguiente enlace:

<http://www.veoh.com/watch/v83500502eS5w32me>

B.4. Comienza la búsqueda

El sistema permanecerá detenido para preparar el entorno de pruebas. Una vez que el entorno esté preparado pulse el botón delantero de la cabeza del robot para comenzar el proceso de búsqueda. En este momento el sistema empezará la búsqueda del objeto y su posterior recogida de forma autónoma. Si nota que el funcionamiento del sistema no es el adecuado o que el robot no encuentra el objeto deseado detenga la ejecución y láncela otra vez si lo desea. Espere a que el robot recoja el objeto.

Apéndice C

Robot NAO

C.1. Lista de motores NAO

En la figura C.1 se muestra la lista de motores disponibles en el robot NAO.

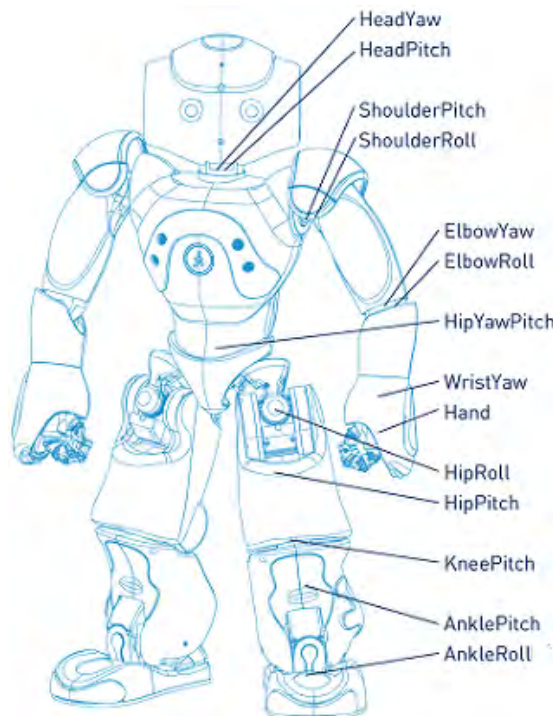


Figura C.1: Motores del robot NAO.

Se puede ver más información de los motores del robot NAO y de su fun-

cionamiento en la página oficial de Aldebaran Robotics https://community.aldebaran-robotics.com/doc/1-14/family/nao_h25/motors_h25.html

C.2. Posición base o inicial

La posición base o inicial es la posición usada de referencia en algunas partes del trabajo. Esta posición se encuentra en *Choregraphe* y los ángulos que la definen se pueden ver en la tabla C.1. En la figura C.2 se puede ver el robot NAO adoptando la posición inicial.

Motor	Valor	Motor	Valor
Cabeza		Piernas	
HeadYaw	0.0	RHipYawPitch	0.0
HeadPitch	0.0	RHipRoll	0.0
Brazos		RHipPitch	-25.0
RShoulderPitch	80.0	RKneePitch	40.0
RShoulderRoll	-20.0	RAnklePitch	-20.0
RElbowYaw	80.0	RAnkleRoll	0.0
RElbowRoll	60.0	LHipYawPitch	0.0
LShoulderPitch	80.0	LHipRoll	0.0
LShoulderRoll	20.0	LHipPitch	-25.0
LElbowYaw	-80.0	LKneePitch	40.0
LElbowRoll	-60.0	LAnklePitch	-20.0
Manos		LAnkleRoll	0.0
RWristYaw	0.0		
RHandl	0.0		
LWristYaw	0.0		
LHandl	0.0		

Tabla C.1: Posición inicial del robot NAO.

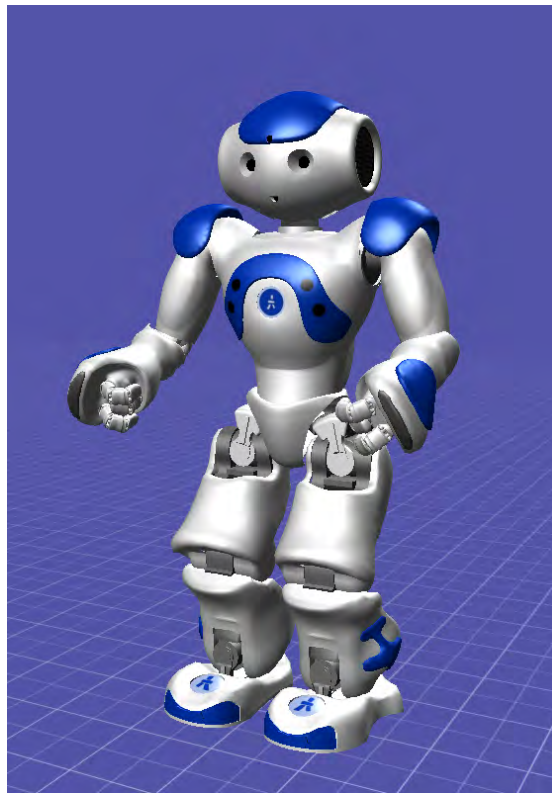


Figura C.2: Posición inicial del robot NAO.

Bibliografía

- [1] “Web oficial de la real academia española www.rae.es.” En línea, ultima comprobación 18 de febrero de 2015.
- [2] “The robotics institute (ri) www.ri.cmu.edu.” En línea, ultima comprobación 18 de febrero de 2015.
- [3] K. Capek, *Rossum’s Universal Robots*. Science-fiction story, 1921.
- [4] J. Freedman, *Robots Through History*. Robotics (New York, N.Y.), Rosen Publishing Group, Incorporated, 2011.
- [5] A. Rajvanshi, “Nikola tesla — the creator of the electric age,” *Resonance*, vol. 12, no. 3, pp. 4–12, 2007.
- [6] J.-H. Kim, S.-H. Choi, I.-W. Park, and S. Zaheer, “Intelligence technology for robots that think [application notes],” *Computational Intelligence Magazine, IEEE*, vol. 8, pp. 70–84, Aug 2013.
- [7] J. Devol, “Programmed article transfer,” June 13 1961. US Patent 2,988,237.
- [8] “Sri international’s ”shakey the robot”selected as robot hall of fame inductee,” Jul 12 2004. Nombre - SRI International; Carnegie Mellon University; Carnegie Science Center; Copyright - Copyright Business Wire Jul 12, 2004; Última actualización - 2012-09-17.
- [9] C. Santos and M. Ferreira, “Timed trajectory generation using dynamical systems: Application to a puma arm,” *Robotics and Autonomous Systems*,

- vol. 57, no. 2, pp. 182 – 193, 2009. Selected papers from 9th International Conference on Intelligent Autonomous Systems (IAS-9) 9th International Conference on Intelligent Autonomous Systems.
- [10] “Another bad break for dante,” Jan 15 1993.
- [11] D. Wettergreen, H. Pangels, and J. Bares, “Behavior-based gait execution for the dante ii walking robot,” in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 3, pp. 274–279 vol.3, Aug 1995.
- [12] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, “The intelligent asimo: system overview and integration,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2478–2483 vol.3, 2002.
- [13] T. S. Kuhn, *The structure of scientific revolutions*. Chicago: University of Chicago Press, 1970.
- [14] I. Linkov, *Real-Time and Deliberative Decision Making (NATO Science for Peace and Security Series C: Environmental Security)*. Springer, 1 ed., 2008.
- [15] R. Brooks, “A robust layered control system for a mobile robot,” *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.
- [16] L. De Silva and H. Ekanayake, “Behavior-based robotics and the reactive paradigm a survey,” in *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*, pp. 36–43, Dec 2008.
- [17] R. C. Arkin, “Integrating behavioral, perceptual, and world knowledge in reactive navigation,” *Robot. Auton. Syst.*, vol. 6, pp. 105–122, June 1990.
- [18] “Web oficial de aldebaran robotics www.aldebaran.com.” En línea, ultima comprobación 18 de febrero de 2015.
- [19] J. A. S. Sánchez, *Avances en robótica y visión por computador*. 2002.

- [20] R. G. Kuehni, *Color : an introduction to practice and principles*. 2012.
- [21] “Resolución de 29 de diciembre de 2013, de la dirección general de empleo, por la que se registra y publica el convenio colectivo del grupo t-systems.” BOE-S-2014-19 <http://www.boe.es/boe/dias/2014/01/22/pdfs/BOE-A-2014-639.pdf>.